

Advancing Vision Without Human Supervision Through Multimodal and Unsupervised Learning

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Dipl.-Inform. Nina Shvetsova
aus Kursk/Russland

Tübingen
2025

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:

08.05.2026

Dekan:

Prof. Dr. Thilo Stehle

1. Berichterstatterin:

Prof. Dr. Hilde Kuehne

2. Berichterstatter:

Prof. Dr. Matthias Bethge

3. Berichterstatter:

Prof. Dr. Yuki M. Asano

ABSTRACT

Building robust, generalizable models that work in the wild across many applications requires learning that does not rely on manual labels. Human annotations are costly and time-consuming. Labels inherit human biases and conventions, reflect taxonomic choices and annotation artifacts, and cannot be obtained at scale or keep pace with an always-changing, long-tailed world. Meanwhile, vast unlabeled data across many modalities (e.g., video, audio, text) are freely available online. Humans naturally learn without explicit teaching by discovering regularities in unimodal and multimodal inputs, integrating information across modalities, and using language to reason about them. This thesis advances a label-free path for image and video understanding, building on ideas of multimodal learning, reasoning in language space, and unsupervised learning.

This thesis contributes to different main stages of building vision models without labels: from creating high-quality datasets for pretraining, to training strong general-purpose models, adapting them to specific tasks and domains, exploring models’ dense properties, and finally improving evaluation benchmarks – all without any human annotations. We develop three complementary ideas: (1) leverage multimodal pretraining to go beyond simple modality-alignment by learning fused multimodal representations, enabling dense spatio-temporal grounding capabilities, and using multimodal models as pseudo-supervisors for task/domain adaptation; (2) use language as the powerful operating space by translating visual signals into text descriptions and exploiting the strong reasoning abilities of large language models; and (3) design losses, architectures, and pipelines for unsupervised learning across different stages, such as pretraining, domain/task adaptation, and spatio-temporal grounding. Concretely, we introduce an ordering-based contrastive objective for self-supervised training that improves the local neighborhood structure of the learned embedding space compared to a commonly used contrastive loss. We then construct truly multimodal video–audio–text representations that can flexibly fuse any subset of modalities into a single joint embedding. Then, we present a pipeline that adapts image–language models to activity-centric video understanding, namely action recognition, using only unlabeled videos together with action dictionaries, eliminating the need for paired supervision. We also develop a pipeline that adapts vision–language models to specific domains (e.g., cooking) using only domain text data, without even target-domain vision data or any paired annotations. To increase the quality of web-scale video–audio–text data for multimodal pretraining without human curation, we propose a method to convert noisy subtitles of web instructional videos into dense, high-quality, human-like descriptions by using LLM to reason about the videos entirely in language space. Using the same language-space reasoning idea, we propose a method to diagnose and mitigate representation biases, such as object or single-frame bias, in popular video evaluation benchmarks, enabling a more robust assessment of video understanding capabilities. We then propose an architecture that learns spatio-temporal activity localization without any grounding supervision, using only web multimodal video–text data. Finally, we unlock activity-grounding abilities in pretrained multimodal models entirely without training, via lightweight architectural modifications.

Together, these contributions advance image and video understanding without reliance on manual labels. We develop methods for label-free model pretraining and adaptation, reveal dense grounding capabilities without human annotations, and introduce unsupervised tools that improve pretraining data quality and create fairer, more reliable evaluation benchmarks.

ZUSAMMENFASSUNG

Der Aufbau robuster, generalisierbarer Modelle, die in der Praxis über viele Anwendungen hinweg funktionieren, erfordert Lernverfahren, die nicht auf manuelle Beschriftungen angewiesen sind. Menschliche Annotationen sind kostspielig und zeitaufwendig, können Annotationsfehler beinhalten und können weder in großem Maßstab erstellt werden noch mit einer sich ständig wandelnden Welt Schritt halten. Gleichzeitig stehen im Internet große Mengen unbeschrifteter Daten über viele Modalitäten hinweg (z. B. Video, Audio, Text) zur Verfügung. Diese Dissertation verfolgt einen label-freien Ansatz für das Lernen von Bild- und Videorepräsentationen, der auf den Konzepten des multimodalen und des unüberwachten Lernens aufbaut.

Diese Arbeit betrachtet dabei verschiedene Phasen des multimodal, unüberwachten Lernens: von der Erstellung hochwertiger Datensätze für das Vortraining über das Training grosser, allgemein einsetzbarer Modelle, deren Anpassung an spezifische Aufgaben und Domänen, bis hin zur Verbesserung von Evaluationsbenchmarks. Sie basiert auf drei komplementären Ideen: (1) der Nutzung eines multimodalen Vortrainings, in dem fusionierte multimodale Repräsentationen gelernt werden, und der Nutzung multimodaler Modelle als Pseudo-Supervisoren für zur Anpassung an bestimmte Aufgaben oder Domänen; (2) der Nutzung von Sprache als leistungsfähigem Repräsentationsraum, indem visuelle Signale in Textbeschreibungen übersetzt und die Schlussfolgerungsfähigkeiten großer Sprachmodelle ausgenutzt werden; und (3) der Anpassung von Loss-Funktionen, Architekturen und Pipelines für unüberwachtes Lernen in verschiedenen Phasen. Konkret führen wir einen Sorting-Loss für das selbstüberwachte Training ein, welcher die lokale Nachbarschaftsstruktur eines Repräsentationsraums im Vergleich zum häufig verwendeten Contrastive Loss verbessert. Anschließend konstruieren wir multimodale Video-Audio-Text-Repräsentationen, die flexibel jede Teilmenge von Modalitäten in einer gemeinsamen Repräsentationen fusionieren können. Daraufhin wird eine Pipeline vorgestellt, die Vision-Language-Modelle für das Videoverstehen – insbesondere für die Bewegungsklassifikation – anpasst, indem ausschließlich unbeschriftete Videos zusammen mit entsprechenden Labels, aber ohne direkte Zuordnung, verwendet werden, wodurch der Bedarf an paarweiser Annotation entfällt. Des weiteren wurde eine Pipeline entwickelt, die Vision-Language-Modelle an spezifische Domänen (z. B. Kochen) anpasst, wobei nur domänenspezifische Textdaten genutzt werden – ohne jegliche Zieldomänen-Bilddaten oder gepaarte Annotationen. Um die Qualität von Video-Audio-Text-Daten im Web-Maßstab für das unüberwachte, multimodale Training zu erhöhen, wurde eine Methode entwickelt, mit der verrauschte Untertitel von webbasierten Videos in hochwertige Beschreibungen umgewandelt werden, indem ein großes Sprachmodell das Video ausschließlich im Sprachraum interpretiert. Mit derselben Idee der sprachbasierten Verarbeitung wurde eine Methode zur Diagnose von visuellen Biases – etwa Objekt- oder Einzelframe-Bias – in gängigen Video-Evaluationsbenchmarks entwickelt, um eine robustere Bewertung aktueller Videomodelle zu ermöglichen. Schliesslich wird eine Architektur vorgestellt, die räumlich-zeitliche Aktivitätslokalisierung unter Verwendung ausschließlich webbasierter multimodaler Video-Text-Daten und ohne jegliche lokale Annotation lernt. Diese wird schliesslich ergänzt durch eine Methode, die Aktivitätslokalisierung auch durch vortrainierte multimodale Modelle ohne weiteres Training lediglich durch architektonische Modifikationen erlaubt.

Insgesamt ermöglichen die genannten Beiträge das Training von Modellen zum Bild- und Videoverstehen ohne die Notwendigkeit manueller Beschriftungen zu verbessern. Die entwick-

elten Methoden für label-freies Training zeigen dabei, dass es möglich ist, gute Repräsentationsfähigkeiten ohne menschliche Annotationen zu lernen und stellen Werkzeuge vor, die die Qualität der Trainingsdaten verbessern und verlässlichere Evaluationsbenchmarks schaffen.

ACKNOWLEDGEMENTS

I would like to express my warmest gratitude to all the people whose guidance, support, and kindness made this PhD journey possible.

First and foremost, I would like to thank my PhD advisor, Hilde Kuehne, who made my PhD such a fulfilling journey. I could not have wished for a more supportive, energetic, and inspiring advisor. She taught me so much that it would be impossible to mention everything: from how to identify important research directions to how to supervise students, but most of all, she showed me that research can be ambitious and joyful at the same time. Her optimism and energy are so contagious, and her belief in people makes those around her want to do better research. Thank you for trusting me with ambitious ideas. Thank you for giving me the freedom to explore. Thank you for always being there with advice when I needed it. I am profoundly grateful for your guidance, your care, and for making my PhD years feel not only productive, but also joyful and alive.

I am also deeply grateful to Bernt Schiele, whose group I joined in the second year of my PhD for what was meant to be a temporary academic stay, but which became one of the most important parts of my PhD: I remained in the group until the end, and Bernt became my second PhD advisor. I am grateful for the amazing collaborative research environment he created, where research discussions were open, thoughtful, and constructive. His perspective helped me see the bigger picture and ask what makes an idea truly significant. It helped me understand research not only as solving problems, but as choosing the right problems to solve. I am thankful for his guidance, his research perspective, and his advice throughout my PhD. It has been a privilege to be part of such a strong and collaborative group.

I would like to sincerely thank Christian Rupprecht, my ELLIS PhD co-advisor, for making my stay in Oxford one of the most memorable parts of my PhD. I learned a lot from the way Christian thinks about research and communicates ideas, and from his remarkable ability to clarify ideas: to see what matters, what can be made simpler, and how a project can be told so that others can see its value. I am extremely grateful for the opportunity to experience PhD life in Oxford, and for the kindness and warmth with which he welcomed me there. This visit gave me a new perspective on research and PhD life, and influenced me deeply.

I further would like to thank my wonderful collaborators at the MIT-IBM Watson AI Lab — Andrew Rouditchenko, Brian Chen, Rogerio S. Feris, James Glass, Leonid Karlinsky, Samuel Thomas, David Harwath, Brian Kingsbury, and Kevin Duarte — with whom I worked on my first PhD paper and continued to collaborate throughout my PhD. This first project showed me, very early on, how powerful collaboration can be, and how much a project can grow through different perspectives and collective effort. I am thankful for the scientific discussions, the shared work, and the inspiration this collaboration gave me for the rest of my PhD.

I am also grateful to my hosts and collaborators during my Google Zurich internship — Goutam Bhat, Prune Truong, and Federico Tombari. Thank you for welcoming me so warmly and for introducing me to an exciting new research direction. Working on XR-related problems gave me a renewed sense of inspiration and curiosity and showed me how rewarding it can be to explore a field where ideas can be experienced so directly.

I am also thankful to all my many amazing collaborators from different institutions and research groups, in particular Anna Kukleva, Felix Petersen, Xudong Hong, Wei Lin, Sirnam Swetha, Arsha Nagrani, Felix Vogel, Walid Bousselham, Siddhartha Gairola, and Sofian

Chaybouti. Working with you taught me that research does not belong to one place: ideas can travel across groups, countries, and time zones, and become stronger through exchange and shared effort. I am grateful for the perspectives each of you brought into my PhD, for the support during different stages of our projects, and for making collaboration feel open and rewarding.

I would also like to thank all members of the D2 group at MPI, VGG Oxford, the CVML group through its time in Frankfurt, Bonn, and finally Tübingen. Thank you for creating such welcoming and inspiring research environments. I am grateful for the discussions, office conversations, group activities, and friendships that grew during these years. Being part of these communities made my PhD richer and much warmer.

To my dearest friends from Saarbrücken, Tübingen, Oxford, Zurich, and far beyond academia, now spread across so many corners of the world — thank you for being my support, my laughter, and my safe place during these years. You were there for the long conversations, the small celebrations, and the days when I needed to be reminded that a paper is not the whole world. You gave me a place to bring my doubts, and helped me return to myself when research became too heavy. You made difficult days easier and good days brighter.

To my parents — long before this thesis, you taught me to be interested in the unknown and not to be afraid of complex problems. I am deeply grateful for nurturing in me curiosity, persistence, and the joy of solving difficult puzzles. To my lovely sister, thank you for your positivity, your support, and for bringing light into many moments when I needed it.

And finally, to my dearest husband, Denis Shvetsov, who has been my superpower throughout the PhD. Throughout these years, you had faith in me and supported what I was doing, however difficult the path became. You were the person I could always lean on: the one who could cheer me up after difficult days, listen to my worries and ideas, and make me feel that I was never alone in this journey. Your patience, love, and belief carried me through more than I can express here. I could not have done this without you, and I am grateful beyond words to have had you by my side.

RELEASED CODE AND DATA

The code and data accompanying this work is publicly available under:

LEARNING BY SORTING:	https://github.com/ninatu/learning_by_sorting
EVERYTHING AT ONCE:	https://github.com/ninatu/everything_at_once
MATCH, EXPAND AND IMPROVE:	https://github.com/wlin-at/MAXI
HowToCAPTION:	https://github.com/ninatu/howtocaption
UNBIASING:	https://github.com/ninatu/utd-project
WHAT, WHEN, AND WHERE:	https://github.com/brian7685/STG
VIDEOGEM:	https://github.com/felixVogelo2/VideoGEM

CONTENTS

1	Introduction	1
1.1	Pretraining Representations Without Human Supervision	4
1.1.1	Contributions	5
1.2	Finetuning Multimodal Models Without Human Supervision	5
1.2.1	Contributions	6
1.3	Improving Data Quality Without Human Supervision	7
1.3.1	Contributions	8
1.4	Grounding with Multimodal Models Without Human Supervision	8
1.4.1	Contributions	9
1.5	Publications	9
2	Related Work	13
2.1	Pretraining Representations	13
2.1.1	Self-supervised Learning	13
2.1.2	Multimodal Pretraining	16
2.1.3	Connection to Our Work	19
2.2	Adapting Representations to Downstream Tasks	19
2.2.1	Action Recognition	20
2.2.2	Text–Video Retrieval	20
2.2.3	Connection to Our Work	21
2.3	Improving Data Quality	21
2.3.1	Improving Large-Scale Multimodal Datasets	22
2.3.2	Debiasing Evaluation Benchmarks	23
2.3.3	Connection to Our Work	23
2.4	Grounding with Multimodal Models	24
2.4.1	Spatial Video Grounding	24
2.4.2	Temporal Video Grounding	25
2.4.3	Spatio-Temporal Video Grounding	25
2.4.4	Connection to Our Work	26
	I Pretraining Representations Without Human Supervision	27
3	Learning by Sorting: Self-supervised Learning with Group Ordering Constraints	29
3.1	Introduction	30
3.2	Related Work	31
3.2.1	Self-supervised Representation Learning	31
3.2.2	Differentiable Sorting and Ranking	33
3.3	Method	33
3.3.1	Training Pipeline	33
3.3.2	Group Ordering Constraints (GroCo)	34
3.3.3	Learning by Sorting	34
3.4	Experimental Evaluation	37
3.4.1	Implementation Details	37

3.4.2	Evaluation Procedure	38
3.4.3	Comparison to State-of-the-Art	38
3.4.4	Comparison to the Pairwise Contrastive Loss	39
3.4.5	Ablation Study	41
3.5	Conclusion	42
4	Everything at Once – Multi-modal Fusion Transformer for Video Retrieval	43
4.1	Introduction	44
4.2	Related Work	46
4.2.1	Multi-modal learning	46
4.2.2	Multi-modal learning with transformers	46
4.3	Method	47
4.3.1	Problem Statement	47
4.3.2	Model Architecture	47
4.3.3	Combinatorial Loss	49
4.4	Experimental Evaluation	50
4.4.1	Experimental Setup	50
4.4.2	Datasets, Tasks, and Metrics	51
4.4.3	Comparison to State-of-the-Art	52
4.4.4	Ablation Study	53
4.5	Conclusion	56
II	Finetuning Multimodal Models Without Human Supervision	57
5	Match, Expand and Improve: Unsupervised Finetuning for Zero-Shot Action Recognition with Language Knowledge	59
5.1	Introduction	60
5.2	Related Work	61
5.2.1	Vision-language Models	61
5.2.2	Zero-Shot Action Recognition	62
5.3	Method	62
5.3.1	CLIP on Video Data	63
5.3.2	Text Bag Construction	64
5.3.3	Multiple Instance Learning	65
5.4	Experimental Evaluation	65
5.4.1	Datasets	65
5.4.2	Implementation Details	66
5.4.3	Zero-Shot Action Recognition	66
5.4.4	Few-Shot Action Recognition	67
5.4.5	Ablation Study	68
5.5	Conclusion	71
6	In-Style: Bridging Text and Uncurated Videos with Style Transfer for Text-Video Retrieval	73
6.1	Introduction	74
6.2	Related Work	75
6.2.1	Text-Video Retrieval	75
6.2.2	Large-scale Multimodal Pretraining	76

6.3	Task: Uncurated & Unpaired Text-Video Retrieval	77
6.4	Method	77
6.4.1	Pseudo Matching	78
6.4.2	Style Transfer	78
6.4.3	Training and Retrieval	79
6.5	Experimental Evaluation	80
6.5.1	Dataset Details	81
6.5.2	Implementation Details	82
6.5.3	Text Query Style	83
6.5.4	Uncurated & Unpaired Text-Video Retrieval	84
6.5.5	Comparison to State-of-the-Art	85
6.5.6	Efficiency of Style Transfer	85
6.5.7	Ablation Study	86
6.6	Conclusion	87
 III Improving Data Quality Without Human Supervision		89
7	HowToCaption: Prompting LLMs to Transform Video Annotations at Scale	91
7.1	Introduction	92
7.2	Related Work	93
7.2.1	Large-Scale Video-Language Datasets	93
7.2.2	Learning with Noisy ASR Subtitles of Instructional Videos	94
7.2.3	LLMs in Vision-Language Tasks	94
7.3	Method	95
7.3.1	Problem Statement	95
7.3.2	Video-Language Retrieval Model	95
7.3.3	HowToCaption Method	96
7.3.4	HowToCaption Dataset	97
7.4	Experimental Evaluation	98
7.4.1	Datasets and Metrics	98
7.4.2	Implementation Details	100
7.4.3	Ablation Study	100
7.4.4	Comparison to State-of-the-Art	102
7.5	Conclusion	103
8	Unbiasing through Textual Descriptions: Mitigating Representation Bias in Video Benchmarks	105
8.1	Introduction	106
8.2	Related Work	108
8.2.1	LLMs for Video Understanding	108
8.3	Discovering Representation Biases	108
8.3.1	Definition: Representation Bias	108
8.3.2	Different Representation Biases	109
8.3.3	Method: Unbiasing through Textual Descriptions (UTD)	110
8.4	Benchmarking Datasets	113
8.4.1	Setup	113
8.4.2	UTD-descriptions	113
8.4.3	Concept Representation Bias	113

8.4.4	Common Sense vs. Dataset Bias	114
8.4.5	Temporal Representation Bias	114
8.4.6	UTD-splits	115
8.5	Benchmarking Video Models	115
8.5.1	Setup	115
8.5.2	Results	116
8.6	Conclusion	117

IV Grounding with Multimodal Models Without Human Supervision 119

9	What, when, and where? Self-Supervised Spatio-Temporal Grounding in Untrimmed Multi-Action Videos from Narrated Instructions	121
9.1	Introduction	122
9.2	Related Work	124
9.2.1	Supervised Spatio-temporal Grounding	124
9.2.2	Multimodal Self-supervised Learning	125
9.3	Method	125
9.3.1	General Setup	125
9.3.2	Representation Guided Frame Sampling	126
9.3.3	Local Representations for Spatial Localization	127
9.3.4	Global Representations For Temporal Localization	128
9.3.5	Inference for Spatio-Temporal Grounding	128
9.4	GroundingYoutube Benchmark	128
9.5	Experimental Evaluation	129
9.5.1	Datasets	129
9.5.2	Baseline Methods	130
9.5.3	Downstream Tasks	130
9.5.4	Comparison to State-of-the-Art	131
9.5.5	Ablation Study	132
9.5.6	Qualitative Results	134
9.6	Conclusion	134
10	VideoGEM: Training-free Action Grounding in Videos	135
10.1	Introduction	136
10.2	Related Work	138
10.2.1	Spatial Video Grounding	138
10.2.2	Training-free Vision-Language Grounding	138
10.3	Method	139
10.3.1	Background (GEM)	139
10.3.2	GEM for Videos	140
10.3.3	Layer Weighting	141
10.3.4	Prompt Decomposition for Action Grounding	142
10.4	Experimental Evaluation	143
10.4.1	Datasets	143
10.4.2	Implementation Details	144
10.4.3	Comparison to State-of-the-Art	144
10.4.4	Ablation Study	145
10.5	Conclusion	148

11 Conclusion and Future Work	149
11.1 Key Ideas and Insights	150
11.2 Summary of Key Conclusions	150
11.2.1 Pretraining Representations Without Human Supervision	151
11.2.2 Finetuning Multimodal Models Without Human Supervision	151
11.2.3 Improving Data Quality Without Human Supervision	152
11.2.4 Grounding with Multimodal Models Without Human Supervision	152
11.3 Future Directions	153
11.3.1 Data: Building Robust Pretraining and Evaluation Corpora	153
11.3.2 Models: Learning Efficient Multimodal Representations	155
11.3.3 Analysis and Adaptation of Learned Representations	157
11.3.4 A Broader View on the Topic	158
Bibliography	159
A Appendix – Learning by Sorting: Self-supervised Learning with Group Ordering Constraints	209
A.1 Discussion of GroCo/Contrastive/Triplet Loss Relations	209
A.2 Additional Experimental Results	211
A.3 Qualitative Analysis of Learned Representation Space	212
A.4 Odd-even Sorting Network	213
A.5 Implementation Details	214
B Appendix – Everything at Once – Multi-modal Fusion Transformer for Video Retrieval	217
B.1 Additional Experimental Evaluation	217
B.2 Qualitative Analysis	219
B.3 Implementation Details	220
C Appendix – Match, Expand and Improve: Unsupervised Finetuning for Zero-Shot Action Recognition with Language Knowledge	225
C.1 Dataset Statistics	225
C.2 Implementation Details	226
C.3 Additional Results	228
C.3.1 Attention Heatmaps	228
C.3.2 Robustness Against Noisy Action Dictionary	230
C.3.3 Examples of Language Sources	231
C.3.4 Parameter-Free Temporal Module	232
D Appendix – In-Style: Bridging Text and Uncurated Videos with Style Transfer for Text-Video Retrieval	233
D.1 Additional Discussions	233
D.2 In-Style Method Details	236
D.3 Implementation Details	237
D.4 Qualitative Results	238
D.5 Limitations	238
E Appendix – HowToCaption: Prompting LLMs to Transform Video Annotations at Scale	243
E.1 Grounding Captions to Video Content with MiniGPT-4	243
E.2 Additional Experimental Evaluation	244

E.2.1	Dataset Quality	244
E.2.2	Additional Comparisons with SOTA in Text-Video Retrieval	245
E.2.3	Additional Results in Prompt Engineering	246
E.2.4	Ablations of Filtering & Alignment Post-processing	246
E.2.5	Robustness Analysis	247
E.3	Additional Implementation Details	247
E.4	HowToCaption Dataset Statistics	249
E.5	Qualitative Examples	250
E.6	Limitations	251
E.7	Responsibility to Human Subjects	251
F	Appendix – Unbiasing through Textual Descriptions: Mitigating Representation Bias in Video Benchmarks	257
F.1	Additional Results	257
F.1.1	Benchmarking Video Models	257
F.1.2	Common Sense Bias	258
F.1.3	Ablation Study	258
F.1.4	Class Distribution in UTD-Splits	260
F.2	Additional Implementation Details	262
F.2.1	Obtaining Textual Descriptions	262
F.2.2	Getting Text Embeddings	262
F.2.3	Unbiasing Datasets	262
F.2.4	Benchmarking Video Models	264
F.3	Qualitative Results	264
F.4	License Information	265
F.5	Limitations and Broader Impact	265
G	Appendix – What, when, and where? Self-Supervised Spatio-Temporal Grounding in Untrimmed Multi-Action Videos from Narrated Instructions	275
G.1	Method Details	275
G.2	Experimental Setup	276
G.2.1	Baseline Details	276
G.2.2	Backbones and Training	276
G.2.3	Inference	277
G.2.4	Evaluation Metrics	278
G.3	Additional Experiments	279
G.3.1	Runtime Analysis	279
G.3.2	Single-Action Spatio-Temporal Grounding	279
G.3.3	Ablation and Decision Choices	280
G.4	GroundingYoutube Annotation	282
G.4.1	Development of the Graphical User Interface and Task Description	282
G.4.2	Quality Control	284
G.4.3	Dataset Usage for Evaluation	285
H	Appendix – VideoGEM: Training-free Action Grounding in Videos	287
H.1	Weighted GEM vs. GEM	287
H.2	Additional Experimental Evaluation	288
H.2.1	Additional Backbone	288
H.2.2	Video Input for Image-Language Models	289

H.2.3	Ablation for Static and Dynamic Weights	290
H.2.4	Ablation for Weight Coefficients	291
H.2.5	Comparison to Multimodal LLMs	291
H.2.6	Ablation for CLIP Prompting	293
H.3	Qualitative Analysis	293
H.4	Limitations	296

INTRODUCTION

Contents

1.1	Pretraining Representations Without Human Supervision	4
1.1.1	Contributions	5
1.2	Finetuning Multimodal Models Without Human Supervision	5
1.2.1	Contributions	6
1.3	Improving Data Quality Without Human Supervision	7
1.3.1	Contributions	8
1.4	Grounding with Multimodal Models Without Human Supervision	8
1.4.1	Contributions	9
1.5	Publications	9

The success of deep learning in computer vision has traditionally relied on the availability of human-annotated datasets. Landmark benchmarks such as ImageNet [DDS⁺09] or Kinetics [KCS⁺17] have driven progress by providing millions of labeled examples, enabling supervised models to learn powerful representations. However, constructing such datasets requires extensive human effort, is expensive, and scales poorly, especially for domains like video, where annotations must also capture temporal boundaries of actions or events. Moreover, human labels are often noisy, incomplete, and biased, reflecting annotation artifacts rather than the underlying structure of the data [TE11, PRB⁺21]. Beyond errors, the very process of defining annotation guidelines inherently constrains what is captured: deciding on label taxonomies or descriptive granularity reflects human assumptions and conventions rather than objective truth [BSoo]. These limitations motivate a shift toward *multimodal and unsupervised learning*, which leverage signals abundantly available online, such as videos with subtitles, images with alt-text, audio–visual correlations, or intrinsic structure of unimodal raw data.

Humans perceive their environment by integrating information from *various sensory modalities*. Vision, hearing, touch, and smell are not processed in isolation but combined into a coherent understanding of the world [GSo6]. For example, watching a speaker’s lip movements improves speech understanding in noisy environments compared to listening alone [SP54, GSo0]. Similarly, training people to visually discriminate motion patterns while concurrently hearing congruent audio leads to faster learning of a visual-only task than training with vision alone [KSSo8, SSo8]. This shows that different modalities can *complement and supervise* each other, enabling more robust understanding of the world and faster learning [MDFB⁺21, BLoo, SSo8]. *Multimodal learning* in computer vision tries to build on those mechanisms and utilize multiple modalities to enable a more reliable, human-like perception and foster more efficient learning of visual representations.

Additionally, beyond complementing and supervising each other, *modalities can also be transferred* from one modality into another. For example, people summarize movies into short plot descriptions or tag lists to allow for easier search and recommendation rather than relying on short video trailers. Similarly, people often understand a visual task better when it is not only explained through visual examples but when this visual task is described in words [May05]. Recent advances in large language models (LLMs) excel at zero-shot reasoning and leveraging background knowledge [BMR⁺20, AAA⁺23]. Systems that *translate vision*

perception into language can improve visual understanding by expressing visual information as text and allowing LLMs to reason about it and/or process this information in the context of their world knowledge [ZLI⁺24].

Finally, humans acquire knowledge largely without explicit supervision. No one provides infants with exhaustive labels of every object, boundary, or action they encounter. Instead, humans learn by *observing and discovering regularities* in sensory input [SAN96, FA02, FA01]. Children do not learn to distinguish an apple from a ball through annotated datasets, but rather by repeatedly encountering them across varied contexts. This ability to learn from raw, uncurated experience is also reflected in the power of *unsupervised learning* in computer vision.

A *central goal* of this thesis is to advance image and video understanding without relying on human annotations, by drawing inspiration from how people learn through (1) multimodal experience, (2) language reasoning, and (3) finding regularities in raw input. **We leverage multimodal data** to enrich visual representations beyond vision alone and to enable modalities to supervise each other, **we transform modalities into language space** to take advantage of the structured reasoning abilities of large language models, and **we adopt unsupervised learning paradigms** to learn directly from unannotated signals. Concretely, this thesis covers the different stages of building models for image and video understanding: from creating high-quality multimodal datasets for pretraining, to developing strong general-purpose models, adapting them to specific tasks and domains, exploring their dense properties, and finally debiasing evaluation benchmarks – all without any human annotations.

The thesis is structured into four main parts.

- **In Part I, Pretraining Representations Without Human Supervision**, we investigate learning strong, generalizable representations from large-scale, annotation-free data that transfer to diverse downstream tasks.
 - In Chapter 3, we propose *a new loss function* that extends the contrastive loss commonly used in self-supervised and multimodal pretraining by relaxing the objective to enforce *only* that positives lie closer to the anchor than any negatives. Using our new objective, we train image representations from raw images and show that it yields more *coherent local neighborhoods* in the embedding space than contrastive loss, resulting in significantly improved performance across benchmarks. This work, “Learning by Sorting: Self-supervised Learning with Group Ordering Constraints”, is published at ICCV 2023 [SPK⁺23].
 - Further, in Chapter 4, building on the idea of multimodal pretraining, where one modality supervises another in a contrastive way using internet data (e.g., CLIP [RKH⁺21a]), we focus on learning a representation that captures not only individual modalities but also any of their combinations, e.g., video-audio-text, in a single *fused* embedding, building truly multimodal representations that exploit *complementary information* and training using only freely available, noisy instructional videos. This work, “Everything at Once – Multi-modal Fusion Transformer for Video Retrieval” is published at CVPR 2022. [SCR⁺22],
- **In Part II, Finetuning Multimodal Models Without Human Supervision**, we address the challenge of adapting pretrained multimodal models to specific tasks or domains without labeled data. While pretrained generalized representations usually achieve strong zero-shot performance, finetuning on supervised datasets often remains crucial for unlocking their full potential and achieving state-of-the-art results in a particular task/domain.

-
- In particular, in Chapter 5, we investigate how to *adapt image–language models*, which tend to focus primarily on objects, *toward video understanding* with a focus on *activities* without relying on any paired human annotations using only unlabeled videos and action dictionaries. This work, “Match, Expand and Improve: Unsupervised Fine-tuning for Zero-Shot Action Recognition with Language Knowledge”, is published at ICCV 2023 [LKS⁺23].
 - In Chapter 6, we further explore how to adapt vision–language representations *to a specific domain* (e.g., cooking) — again without any paired annotations, moreover, without even target-domain video data and using only target-domain language. This work, “In-Style: Bridging Text and Uncurated Videos with Style Transfer for Text-Video Retrieval”, is published at ICCV 2023 [SKSK23].
 - Further, in **Part III, Improving Data Quality Without Human Supervision**, we focus on improving the quality of training and evaluation data without human involvement. For multimodal pretraining of video models, natural multimodal signals such as instructional videos paired with dense narrations obtained via automatic speech recognition are freely available online. However, these narrations are often very noisy and misaligned and thus suboptimal for effective pretraining. In addition, existing video evaluation benchmarks are also limited, as they frequently exhibit representation biases, such as object or single-frame bias, where accurate predictions can be achieved merely by recognizing objects or relying on a single frame.
 - Therefore, in Chapter 7, we show how to *generate high-quality dense video annotations* at scale without any human supervision from subtitles obtained from automatic speech recognition by formulating and solving the problem in language space and using LLMs for reasoning. Using this method, we create a new video–audio–text dataset, *HowToCaption*, which yields substantially better downstream performance on video–audio–text tasks compared to the original noisy supervision. This work, “HowToCaption: Prompting LLMs to Transform Video Annotations at Scale”, is published at ECCV 2024 [SKH⁺24].
 - Further, in Chapter 8, we also show that mapping videos to language space allows us to estimate and mitigate different types of *representation biases* in video evaluation benchmarks at scale without any human input. We conduct a systematic analysis of popular video benchmarks and create new debiased test splits, which we believe provide a more robust assessment of video understanding capabilities. This work, “Unbiasing through Textual Descriptions: Mitigating Representation Bias in Video Benchmarks”, is published at CVPR 2025 [SNS⁺25].
 - Finally, in **Part IV, Grounding with Multimodal Models Without Human Supervision**, we address dense vision tasks, specifically the localization of actions and events in space and time in videos without any grounding supervision during training.
 - In Chapter 9, we explore how *multimodal signals from instructional videos* can guide *spatio-temporal grounding of activities* in untrimmed videos. We also propose a new evaluation benchmark with dense spatio-temporal grounding annotations in long, untrimmed, multi-action instructional videos, enabling assessment of this challenging task in a real-world setting. This work, “What, when, and where? Self-Supervised Spatio-Temporal Grounding in Untrimmed Multi-Action Videos from Narrated Instructions”, is published at CVPR 2024 [CSR⁺24].

- In Chapter 10, we explore a *training-free* approach that uncovers the inherent *action grounding capabilities* of frozen pretrained multimodal models without any additional training or finetuning. This work, “VideoGEM: Training-free Action Grounding in Videos”, is published at CVPR 2025 [VBK⁺25].

For the rest of this chapter, we introduce each part in more detail in Sections 1.1 to 1.4 and then list all corresponding publications and collaborations in Section 1.5.

1.1 PRETRAINING REPRESENTATIONS WITHOUT HUMAN SUPERVISION

To train powerful, robust models that generalize across diverse tasks, large and diverse datasets are essential. Since manual annotation at such scales is costly, self-supervised learning has emerged as an attractive alternative, offering a paradigm for learning transferable representations from unlabeled data. Self-supervised learning methods include *discriminative approaches*, which learn by solving surrogate tasks such as rotation prediction [GSK18], jigsaw puzzles [NF16], or temporal order prediction in videos [MZH16a], and *generative approaches*, which learn by reconstructing missing or corrupted input, including autoencoders [VLBMo8, VLL⁺10], inpainting [PKD⁺16], colorization [ZIE16, LMS16], or masked token prediction [HCX⁺22, BDPW21, ZWW⁺21]. Discriminative methods also include *contrastive approaches* [DAT⁺21, CKNH20, TSP⁺20, HFW⁺20a, BHB19, YZYC19], which significantly narrowed the gap to supervised learning by training models to pull together positive pairs (e.g., augmented views of the same sample) and push apart negatives (e.g., different samples) in the embedding space. Contrastive objective has proven highly effective not only for unimodal representation learning but also as the foundation of multimodal approaches aligning images, video, audio, and text [RKH⁺21a, MZA⁺19, RBH⁺21] and has been extended in many frameworks [GSA⁺20, CMM⁺20, ZJM⁺21, BPL21, ESSS21, CTM⁺21, CH21b]. Despite its success, contrastive loss has several drawbacks. One drawback is that it **minimizes the distance between positives and maximizes separation from negatives**, which can be suboptimal: positives may still differ despite sharing a concept, and even distant negatives contribute unnecessarily to the optimization. Moreover, **the contrastive objective is pairwise**, aligning only individual positive pairs and negative pairs rather than groups of related positives and negatives. Extensions such as SwAV [CMM⁺20], which incorporates clustering, or Whiten-ing [ESSS21], which aligns multiple positives, address this partially. In this thesis, we build on contrastive learning and propose a new loss that explicitly addresses these limitations.

In addition to self-supervised learning from unimodal data, multimodal pretraining provide another powerful paradigm, relying instead on the *naturally co-occurring signals* found in web data. The key idea is to exploit noisy yet abundant internet-scale resources, such as image and alt-text pairs or video with textual narration obtained from automatic speech recognition, as supervisory signals. Models are trained to project different modalities into a shared embedding space using contrastive objectives, minimizing the distance between positive pairs (e.g., an image and its caption) while pushing apart unrelated samples (e.g., an image and another image’s caption). Powerful image-text [RKH⁺21a, LLXH22a, SWY⁺24, CLW⁺25, ZMKB23], video-text [BNVZ21a, MZA⁺19, ZLH⁺21a, WLL⁺24], and video-audio-text models [RBH⁺21, AYQ⁺21] demonstrate the effectiveness of this paradigm. Such multimodal pretraining demonstrates strong zero-shot transfer to other tasks, such as zero-shot classification or retrieval [RKH⁺21a, LLXH22a, MZA⁺19, RBH⁺21]. Additionally, pretrained encoders are also used as a strong initialization for downstream tasks. However, most efforts have learned a joint embedding space that projects **only individual modalities**, while the challenge

of **jointly encoding multiple modalities**, for example, obtaining a fused embedding for video with accompanying audio and text, in an annotation-free multimodal manner has remained unexplored. Such representations could be particularly valuable for capturing richer multimodal signals and for improving performance on tasks that require reasoning across modalities, for example, classifying videos together with their audio and accompanying text.

1.1.1 Contributions

In this thesis, we make two contributions to this field.

- First, in Chapter 3, we delve into contrastive objectives and propose a new contrastive loss, Group-Ordering Constraints. Instead of comparing pairs of positives and negatives, we **compare groups of positives and groups of negatives** [SPK⁺23]. Moreover, we relax the standard constraint of “pull positives together, push negatives apart” and enforce *only* that **all positive samples are closer to the anchor than all negatives**. This yields a more holistic optimization of local neighborhoods than the standard contrastive loss. We evaluate our new objective for self-supervised learning on unlabeled images and show that it produces representations with more consistent local neighborhoods, evidenced by significantly improved k -NN performance across benchmarks.
- Then, in Chapter 4, we aim to learn multimodal representations enabling not only strong **single-modality representations** but also a **fused representation for arbitrary modality inputs** using only free supervision from naturally co-occurring signals in instructional web videos. We introduce a modality-agnostic fusion transformer that routes information across video, audio, and text to produce a joint embedding [SCR⁺22]. We training the representation with a contrastive objective applied in a combinatorial manner, pulling together any combinations of given modalities. The resulting representations support flexible modality combinations at test time and outperform previous methods in zero-shot performance on retrieval and action localization.

1.2 FINETUNING MULTIMODAL MODELS WITHOUT HUMAN SUPERVISION

Large-scale vision-language models such as CLIP [RKH⁺21a], and SigLIP [ZMKB23] or Meta CLIP [XXT⁺23, CLW⁺25] have achieved remarkable success by training on large-scale corpora of web-scraped image-text pairs. They enable open-vocabulary recognition through zero-shot transfer, powering tasks from classification to detection and segmentation [ZFZ⁺22, ZYLL22a, GLKC22, RZC⁺22]. Yet, despite their strong generalization, most systems still require finetuning on downstream datasets to match fully supervised performance. In video understanding, such datasets are particularly costly to construct because they also require temporal annotation, such as identifying salient segments within long videos, precisely trimming them, and aligning each segment with an appropriate textual description.

One of the main challenges in **adapting image-text models to video understanding**. Image-text models usually are trained on much larger and diverse datasets than video-text models, partly because video data has higher storage and compute costs, noisier and less aligned textual signals, and more restrictive legal and licensing constraints. But despite image-text models is promising for video understanding, they usually underperform on zero-shot video tasks [WXL21, NPC⁺22, JHZ⁺22, WSO23, CH22, RKM⁺22]. This gap is usually attributed to a core bias: these models predominantly capture objects (nouns) while failing to

represent actions (verbs or verb phrases) [TJB⁺22, ZZZ⁺22, YBK⁺23a, HN21]. To overcome this, some methods adapt image-language models for video understanding by equipping them with temporal modeling and applying supervised finetuning [GVM⁺22, LXX⁺22, WXL21, RKM⁺22]. However, such methods still **rely on costly annotations** and **remain constrained by the narrow vocabularies** of curated datasets, limiting zero-shot transfer to unseen categories. In this thesis, we tackle the challenge of adapting image-language models to video understanding using unlabeled video data with an **open vocabulary** in an **unsupervised** manner. While more recent models also solve this problem by combining larger-scale image-text data with video-text data for pretraining [WHL⁺23, WLL⁺24] or by using image-text models as teachers to train video models [LWL⁺23, LLW⁺24], our method enables finetuning in a relatively low-resource regime of strong image-text models and might applied to newer CLIP-family models such as EVA-CLIP-18B [SWY⁺24] and MetaCLIP2 [CLW⁺25].

Another challenge lies in the **adaptation of multimodal models to specific video domains** (e.g., cooking or movies). Most multimodal models utilize supervised finetuning for each domain [RKM⁺22, LWL⁺23, WHL⁺23, WLL⁺24]. But since collecting domain-specific annotations is expensive, some methods attempt to augment domain-specific data automatically [WLF⁺23, ZMKG22] and finetune the model using these data. For example, Cap4Video [WLF⁺23] generates auxiliary captions for in-domain videos by optimizing language model output to maximize CLIP [RKH⁺21b] similarity score to the video. LaViLa [ZMKG22] proposes generating additional narrations for dense coverage of long videos by finetuning a large language model on subset of paired text-video data. However, zero-shot captioning methods such as Cap4Video require **substantial optimization time** and access to in-domain videos, and methods that finetune language models to produce in-domain captions, such as LaViLa, **still rely on partially annotated data**. In this thesis, we make a first step towards finetuning vision-language models using only in-domain text data, without any paired data or even without in-domain videos.

1.2.1 Contributions

This thesis investigates annotation-free finetuning strategies and makes two contributions to the field.

- In Chapter 5, we propose MAtch, eXpand and Improve (MAXI) [LKS⁺23], an approach to **adapt image-text models for open-vocabulary video understanding without requiring annotated data**. MAXI enables finetuning on unlabeled video datasets (e.g., Kinetics-400 [KCS⁺17]) by leveraging external language sources such as action dictionaries, world knowledge of large language models (e.g., GPT-3 [BMR⁺20]), and vision-language models for captioning (e.g., BLIP [LLXH22a]) and pseudo-matching (e.g., CLIP [RKH⁺21a]). We show that MAXI effectively leverages unlabeled video data, significantly improving source model performance for video understanding, competing with state-of-the-art supervised methods, and even surpassing them in some zero-shot and few-shot action recognition tasks.
- In Chapter 6, we introduce a method to **adapt vision-language models for domain-specific text-video retrieval using only text queries from the target domain** [SKSK23]. We formalize this new setup as *text-video retrieval with uncurated and unpaired data*, where training relies solely on text queries without corresponding videos. This setting is motivated by the ease of collecting or generating text data, e.g., producing topic-specific queries rather than curating matched video-text pairs. We also propose In-Style method,

which learns the style of domain-specific queries and transfers it to uncured web videos, enabling training on pseudo pairs without any human annotations and significantly narrowing the gap between zero-shot and fully supervised performance.

1.3 IMPROVING DATA QUALITY WITHOUT HUMAN SUPERVISION

The rapid progress of multimodal pretraining has been fueled by large-scale web data collections for training, such as LAION [SVB⁺21a], DataComp [GIF⁺23], HowTo100M [MZA⁺19], and WebVid [BNVZ21a], which provide massive amounts of image-text or video-text pairs at virtually no annotation cost. While the scale of training data has been the main driver of recent progress, noisy or biased data might limit generalization and robustness. This has sparked growing interest in **automatically improving training data quality** through filtering, denoising, and re-annotation strategies [GIF⁺23, LLXH22a, LAI24, SVB⁺21a, WHL⁺23]. While image-alt-text pairs scraped from the web have proven effective for training image models, especially combined with dataset cleaning strategies [GIF⁺23, XXT⁺23, SVB⁺21a], video-text data introduces unique challenges. Web videos with alt-text, such as those in WebVid [BNVZ21a], **often lack audio and provide only high-level descriptions** rather than dense annotations of multiple events. Instructional videos, e.g. in HowTo100M [MZA⁺19], with narration automatically transcribed by speech recognition systems (ASR) offer a promising alternative, since they supply video-audio-text signals at scale. However, such data is inherently noisy: ASR transcripts are often **ungrammatical and filled with filler words and often weakly aligned with visual content** [MZA⁺19, HXZ22]. While some works mitigate this issue by re-annotating video data with pretrained image captioning models applied to individual frames [WHL⁺23, CSM⁺24, CLW⁺24] – essentially transferring knowledge from images to videos – in this thesis we instead convert noisy ASR transcripts into high-quality, temporally aligned captions, thereby uncovering new knowledge within the data, enabling the construction of densely annotated video-audio-text datasets for more effective pretraining.

Another critical aspect of data quality lies in the evaluation benchmarks used to assess robustness and generalization. The quality of evaluation benchmarks is crucial, as they may suffer from **different type of biases and spurious correlations** that unintentionally hinder robust image and video understanding evaluation. In video understanding, where action classification and text-to-video retrieval datasets are typically used to evaluate the fundamental ability of new models, there is ongoing debate about how much performance on these benchmarks reflects true action recognition versus reliance on spurious cues such as objects or background context [HSSK16, CWR22, WR21]. This issue, known as **representation bias**, has been examined through strategies such as masking human figures or replacing backgrounds [HSSK16, CWR22] or miming actions in unrelated environments [WR21]. Other works aiming to reduce scene or object bias have collected specialized datasets [LLV18, BSYS⁺21, GEKM⁺17b]. Though such debiased solutions are rarely adopted for benchmarking, either due to artifacts in modified videos or perceived out-of-domain data distributions. Beyond object and scene bias, there was also a discussion about whether current video benchmarks truly evaluate temporal reasoning, since single-frame models often achieve competitive performance with video-based methods [BEG⁺22, LBB23]. While new benchmarks have been proposed to explicitly test temporal understanding [BTS23, MAM23, CTZ⁺24], temporal representation bias in standard datasets remains underexplored. In this thesis, we address this challenge by systematically assessing and mitigating different forms of representation bias in video benchmark, including object/scene and temporal bias.

1.3.1 Contributions

This thesis investigates annotation-free strategies for improving both training data and evaluation benchmarks in video understanding. Our main contributions are:

- In Chapter 7, we present a method to **transform noisy ASR transcripts of instructional videos into high-quality, temporally aligned captions**, enabling the construction of densely annotated **video-audio-text datasets** for large-scale pretraining [SKH⁺24]. We design an LLM prompting strategy that processes longer subtitle contexts, generates human-like captions, and aligns them to videos using subtitle timestamps. Applied to HowTo100M dataset, we produce the large-scale HowToCaption dataset. Our evaluation shows that HowToCaption improves zero-shot text-video retrieval and captioning across benchmarks, and further disentangles narration from audio, boosting performance in text-video-audio tasks.
- In Chapter 8, we introduce a scalable approach to **systematically analyze and mitigate different types of representation bias, such as object or single-frame bias, in existing video benchmarks** using automatically generated textual descriptions, without the need for manual annotations [SNS⁺25]. To this end, we leverage VLMs and LLMs, converting our video task into the problem described with language, which allows us to exploit the strong zero-shot reasoning abilities of LLMs and to draw conclusions about biases in video representation. Using our method, we systematically analyze 12 video datasets, create object-debiased test splits, and benchmark 30 state-of-the-art models to reveal both dataset and model biases. To enable more robust evaluation, we release UTD-descriptions (structured dataset annotations) and UTD-splits (object-debiased test sets).

1.4 GROUNDING WITH MULTIMODAL MODELS WITHOUT HUMAN SUPERVISION

Parts I and II explore pretraining and finetuning for image and video understanding using multimodal and unsupervised learning. However, these chapters focus on models that operate at a coarse level and do not address dense prediction tasks, such as semantic segmentation, object detection, spatial or spatio-temporal grounding of objects and activities. In Part IV, we investigate how to equip models with **dense properties**, namely, focusing on enabling models with **action-grounding** capabilities without manual annotations.

Spatio-temporal grounding of activities in video is particularly challenging. Unlike objects with clear spatial extent, actions emerge from interactions unfolding over time, making them ambiguous, variable, and costly to annotate [JGZ⁺13, GSR⁺18, MS17]. Traditional activities grounding methods rely on dense human supervision, pairing bounding-box trajectories with textual descriptions [YMS⁺22, JyLYM22], which limits scalability and generalization. In contrast, multimodal self-supervised approaches [MAS⁺20, ARS⁺20, AYQ⁺21, CRD⁺21] trained on large-scale video-text corpora such as HowTo100M [MZA⁺19] achieve state-of-the-art results at coarse tasks, including zero-shot temporal segmentation [ZAC⁺19a, TDR⁺19, SCR⁺22], but lack spatial localization abilities. Recent works have explored annotation-free spatial grounding [AKB⁺19, LZZ⁺22, WLL⁺22, ZYZ⁺22, TPS⁺21, CSR⁺24], learning to localize entities from image-caption or video-caption pairs without bounding boxes. Yet, these methods **assume the queried concept is visible** and are not designed to detect whether and when an action occurs in untrimmed, real-world videos. This gap motivates us to extend multimodal

training **toward spatio-temporal grounding**, namely localizing events both spatially and temporally without direct grounding supervision, which we address in this thesis.

While the annotation-free grounding methods discussed above avoid manual labels, they often **require specialized architectures or losses** tailored to spatial or temporal localization. In contrast, a separate line of work explores training-free localization of semantic concepts in images [ZLD22, LWDL23, BPFK24] by adapting vision-language models for object grounding. While effective for object localization in images, action localization in videos is considerably more challenging, as it requires capturing contextual cues beyond object boundaries. Unlike objects with clear boundaries, actions do not have clear outlines, they are context-dependent, often involving interactions between multiple entities across varying timescales. In this thesis, we address this challenge by **investigating training-free spatial activity grounding** in videos leveraging frozen pretrained multimodal models.

1.4.1 Contributions

This thesis explores annotation-free grounding of events in videos and makes the following contributions.

- In Chapter 9, we introduce the first approach for **spatio-temporal grounding** of events in untrimmed videos trained **using only multimodal supervision** from instructional videos, without any human annotations [CSR⁺24]. Our method integrates fine-grained local representations, which capture spatial details, with global representations that encode higher-level temporal and contextual information, combining them into a unified framework. We also propose a new benchmark dataset, GroundingYoutube, a first benchmark with dense spatio-temporal annotations centered around activities in long, untrimmed instructional videos containing multiple actions. Experiments on this dataset and other downstream tasks demonstrate that our approach outperforms existing baselines across spatial, temporal, and spatio-temporal grounding scenarios.
- In Chapter 10, we explore how to enable **spatial grounding of activities** using pretrained multimodal models in a **fully training-free manner** [VBK⁺25]. To this end, we adapt the GEM method [BPFK24], originally developed for grounding objects and referring expressions in images, to the problem of activity grounding in videos. We propose a layer-weighting scheme that prioritizes higher layers, motivated by our analysis showing that actions, as abstract semantic concepts, typically emerge in the higher layers of models. We also introduce a prompt decomposition technique that processes verbs, actions, and objects separately, resulting in more accurate spatial localization of activities. Our fully training-free method outperforms existing supervised state-of-the-art approaches for spatial video grounding.

1.5 PUBLICATIONS

Four parts of this thesis comprise eight chapters (Chapter 3 to 10), that present the core contributions of the thesis. Each of Chapter 3 to Chapter 10 is based on a paper that has undergone peer review and was accepted for publication at a conference. The corresponding chapters closely follow the respective publications, with only minor adjustments and formatting changes. Reuse of the published material follows the author reuse rights and licence conditions of the

respective publishers; in particular, the IEEE/CVF¹ chapters are based on the accepted/CVF Open Access versions, while the ECCV/Springer Nature chapter is reproduced in accordance with the Springer Nature Licence to Publish. Below, we list publications included in this thesis, as well as summarize Nina Shvetsova's contributions in each publication, and acknowledge collaborations. Full bibliographic information is provided in the bibliography.

- **Chapter 3: Learning by Sorting: Self-supervised Learning with Group Ordering Constraints.**

Nina Shvetsova, Felix Petersen, Anna Kukleva, Bernt Schiele, and Hilde Kuehne. In *ICCV 2023*. [SPK⁺23]

The work was conducted by Nina Shvetsova in collaboration with Felix Petersen and Anna Kukleva under the supervision of Bernt Schiele and Hilde Kuehne. The initial idea was conceived jointly by Nina Shvetsova, Felix Petersen, and Hilde Kuehne. As first author, Nina Shvetsova was responsible for the study design, code implementation, and evaluation procedure, and conducted all experiments and analyses and was the main writer of [SPK⁺23]. All authors contributed to scientific discussions and to the text of the final publication.

- **Chapter 4: Everything at Once – Multi-modal Fusion Transformer for Video Retrieval.**

Nina Shvetsova, Brian Chen, Andrew Rouditchenko, Samuel Thomas, Brian Kingsbury, Rogerio S Feris, David Harwath, James Glass, and Hilde Kuehne. In *CVPR 2022*. [SCR⁺22]

This work was conducted in collaboration with the MIT-IBM Watson Lab. Nina Shvetsova, as first author, designed the proposed method, implemented it, conducted all experiments and analyses, and was the main writer of [SCR⁺22]. All authors contributed to scientific discussions and to the text of the final publication. Brian Chen and Andrew Rouditchenko also assisted with setting up the initial codebase.

- **Chapter 5: Match, Expand and Improve: Unsupervised Finetuning for Zero-Shot Action Recognition with Language Knowledge.**

Wei Lin, Leonid Karlinsky, **Nina Shvetsova**, Horst Possegger, Mateusz Kozinski, Rameswar, Panda, Rogerio Feris, Hilde Kuehne, and Horst Bischof. In *ICCV 2023*. [LKS⁺23]

This work was done in collaboration with the MIT-IBM Watson Lab and Graz University of Technology. Wei Lin was the lead author, responsible for the study design, code implementation, and conducting all experiments, as well as being the main writer. Nina Shvetsova and the other co-authors contributed to regular scientific discussions on the project.

- **Chapter 6: In-Style: Bridging Text and Uncurated Videos with Style Transfer for Text-Video Retrieval.**

¹In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of University of Tübingen's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Nina Shvetsova*, Anna Kukleva*, Bernt Schiele, and Hilde Kuehne (*equal contribution). In *ICCV* 2023. [SKSK23]

This project was co-led by Nina Shvetsova and Anna Kukleva as co-first authors, under the supervision of Bernt Schiele and Hilde Kuehne. Nina Shvetsova and Anna Kukleva contributed equally to all stages of the work, including the conception of the idea, method design, implementation, experiments, and writing. All authors participated in scientific discussions and contributed to the final manuscript.

- **Chapter 7: HowToCaption: Prompting LLMs to Transform Video Annotations at Scale.** **Nina Shvetsova***, Anna Kukleva*, Xudong Hong, Christian Rupprecht, Bernt Schiele, and Hilde Kuehne (*equal contribution). In *ECCV* 2024. [SKH⁺24]

This project is done by Nina Shvetsova, Anna Kukleva, and Xudong Hong under the supervision of Christian Rupprecht, Bernt Schiele, and Hilde Kuehne. Nina Shvetsova and Anna Kukleva are co-first authors of this work. The initial idea was conceived by Nina Shvetsova, who was responsible for the study design, code implementation, experiments, and was the main writer of [SKH⁺24]. Anna Kukleva contributed equally as co-first author during the initial stages of the project, before transitioning to her internship. Xudong Hong contributed to the NLP aspects and collected statistics of the generated captions. All authors participated in scientific discussions and contributed to the final manuscript.

- **Chapter 8: Unbiasing through Textual Descriptions: Mitigating Representation Bias in Video Benchmarks.**

Nina Shvetsova, Arsha Nagrani, Bernt Schiele, Hilde Kuehne, and Christian Rupprecht. In *CVPR* 2025. [SNS⁺25]

This work was conducted by Nina Shvetsova under the supervision of Arsha Nagrani, Bernt Schiele, Hilde Kuehne, and Christian Rupprecht. As first author, Nina Shvetsova designed the method, implemented it, conducted all experiments and analyses, and was the main writer of the paper. All authors participated in scientific discussions and contributed to the final manuscript.

- **Chapter 9: What, when, and where? Self-Supervised Spatio-Temporal Grounding in Untrimmed Multi-Action Videos from Narrated Instructions.**

Brian Chen, **Nina Shvetsova**, Andrew Rouditchenko, Daniel Kondermann, Samuel Thomas, Shih-Fu Chang, Rogerio Feris, James Glass, and Hilde Kuehne. In *CVPR* 2024. [CSR⁺24]

This work was conducted in collaboration with Columbia University, the MIT-IBM Watson Lab, and Quality Match GmbH. Brian Chen was the lead author of the paper. Nina Shvetsova contributed to weekly scientific discussions, writing of the paper, and creating the figures.

- **Chapter 10: VideoGEM: Training-free Action Grounding in Videos.**

Felix Vogel, Walid Bousselham, Anna Kukleva, **Nina Shvetsova**, and Hilde Kuehne. In

CVPR 2025. [VBK⁺25]

This paper is based on the master thesis of Felix Vogel. Nina Shvetsova acted as Felix Vogel's supervisor during this project.

Publications not included in the thesis, as they are only loosely related to its topic:

- **M2SVid: End-to-end inpainting and refinement for monocular-to-stereo video conversion.**
Nina Shvetsova, Goutam Bhat, Prune Truong, Hilde Kuehne, and Federico Tombari. *Under review*, 2025. [SBT⁺25]
- **Preserving modality structure improves multi-modal learning.**
Sirnam Swetha, Mamshad Nayeem Rizve, Nina Shvetsova, Hilde Kuehne, and Mubarak Shah. In *ICCV*, 2023. [SRS⁺23]
- **C2KD: Cross-lingual cross-modal knowledge distillation for multilingual text-video retrieval.**
Andrew Rouditchenko, Yung-Sung Chuang, Nina Shvetsova, Samuel Thomas, Rogerio Feris, Brian Kingsbury, Leonid Karlinsky, David Harwath, Hilde Kuehne, and James Glass. In *ICASSP*, 2023. [RCS⁺23]
- **Augmentation learning for semi-supervised classification.**
Tim Frommknecht, Pedro Alves Zipf, Quanfu Fan, Nina Shvetsova, and Hilde Kuehne. In *GCPR*, 2022. [FZF⁺22]
- **Mood 2020: A public benchmark for out-of-distribution detection and localization on medical images.**
David Zimmerer, Peter M Full, Fabian Isensee, Paul Jäger, Tim Adler, Jens Petersen, Gregor Köhler, Tobias Ross, Annika Reinke, Antanas Kascenas, Bjørn Sand Jensen, Alison Q. O'Neil, Jeremy Tan, Benjamin Hou, James Batten, Huaqi Qiu, Bernhard Kainz, Nina Shvetsova, Irina Fedulova, Dmitry V. Dylov, Baolun Yu, Jianyang Zhai, Jingtao Hu, Runxuan Si, Sihang Zhou, Siqi Wang, Xinyang Li, Xuerun Chen, Yang Zhao, Sergio Naval Marimont, Giacomo Tarroni, Victor Saase, Lena Maier-Hein, and Klaus Maier-Hein. *IEEE Transactions on Medical Imaging*, 2022. [ZFI⁺22]
- **Anomaly detection in medical imaging with deep perceptual autoencoders.**
Nina Shvetsova, Bart Bakker, Irina Fedulova, Heinrich Schulz, and Dmitry V Dylov. *IEEE Access*, 2021. [SBF⁺21]

RELATED WORK

Contents

2.1	Pretraining Representations	13
2.1.1	Self-supervised Learning	13
2.1.2	Multimodal Pretraining	16
2.1.3	Connection to Our Work	19
2.2	Adapting Representations to Downstream Tasks	19
2.2.1	Action Recognition	20
2.2.2	Text–Video Retrieval	20
2.2.3	Connection to Our Work	21
2.3	Improving Data Quality	21
2.3.1	Improving Large-Scale Multimodal Datasets	22
2.3.2	Debiasing Evaluation Benchmarks	23
2.3.3	Connection to Our Work	23
2.4	Grounding with Multimodal Models	24
2.4.1	Spatial Video Grounding	24
2.4.2	Temporal Video Grounding	25
2.4.3	Spatio-Temporal Video Grounding	25
2.4.4	Connection to Our Work	26

This chapter reviews the literature relevant to the topic of this thesis and is organized according to its four main parts. We present the most relevant and recent work in this field and relate it to the contributions done in this thesis. However, the chapters of the four main parts also include related work discussions more specific to the respective topics of these chapters.

2.1 PRETRAINING REPRESENTATIONS

In this section, we focus on pretraining robust visual representations without human supervision through self-supervised learning and multimodal pretraining, the two main paradigms for label-free representation learning. We discuss related work in this area in detail, as pretrained models without supervision on large-scale web data serves as the foundation for all parts of this thesis. We also highlight connections to our contributions on pretraining representations, presented in Part I, Pretraining Representations Without Human Supervision.

2.1.1 Self-supervised Learning

Self-supervised learning (SSL) is a fast-growing research field that exploit supervisory signals directly available in the data itself. Instead of relying on human labels, self-supervised learning leverages automatically generated objectives, ranging from early pretext tasks [ZIE16, PKD⁺16] to modern contrastive or masked modeling methods [CXH21a, HCX⁺22], to learn discriminative and transferable representations. In this section, we focus on SSL methods for learning vision representations. It is notable that representations pretrained with self-supervised learning [CKNH20, HFW⁺20b, GSA⁺20, CMM⁺20, CTM⁺21, ZJM⁺21, BPL21, HCX⁺22, ODM⁺23] significantly surpass those obtained with supervised pretraining on standard image benchmarks

such as ImageNet [DDS⁺09], particularly in terms of transfer performance on downstream tasks. SSL further enables learning more powerful and generalizable representations by scaling up data and model capacity [SVS⁺25].

Earlier Pretext Tasks. Early self-supervised learning methods used surrogate objectives, namely *pretext tasks*, such as colorization [ZIE16], inpainting [PKD⁺16], rotation prediction [GSK18], and jigsaw puzzle solving [NF16, CLJ21] to encourage models to learn semantic and spatial structure. Puzzle-solving idea was also recently revived for self-supervised pretraining of transformer architectures [WFW⁺23, ZLL⁺23, AAH⁺25] by dropping positional encodings and training the model to recover them. It was shown that such puzzle-based objectives lead to stronger representations than masked-reconstruction objectives, e.g., MAE [HCX⁺22], especially on dense tasks [WFW⁺23].

Clustering. Clustering-based methods [YPB16, XGF16, DSRB14, LSZU16] learn feature representations by optimizing a clustering objective or by creating pseudo-labels from groups of similar samples. However, the main challenge of such approaches is preventing collapse into a trivial solution in which most images are assigned to one or a few clusters. DeepCluster [CBJD18] was one of the first approaches to scale clustering-based methods for training deep representations on ImageNet-scale data. DeepCluster introduced mechanisms to prevent cluster collapse and proposed to alternate between clustering features to obtain pseudo-labels and training the network to predict these labels. This idea is further extended in DeeperCluster [CBMJ19], which trained representations on uncured long-tail data, and in Online Deep Clustering [ZXL⁺20], which removed the need to alternate between clustering and representation learning by updating cluster centroids and network parameters simultaneously. SeLA [ARV20] further proposed to perform clustering and representation learning with a single loss, and mitigated collapse by maximizing the mutual information between sample indices and labels through an optimal transport formulation solved with the Sinkhorn-Knopp algorithm [Cut13]. SwAV [CMM⁺20] combined online clustering with contrastive learning ideas by using soft cluster assignments from the Sinkhorn-Knopp instead of augmented views of the image in the training objective. DINO [CTM⁺21] adopted clustering ideas in self-distillation objective, using with a student-teacher architecture, where the student learns to match the teacher “class” predictions using centering and sharpening to prevent collapse. iBOT [ZWW⁺21] further extended this idea to patch-level self-distillation. Later methods, such as MSN [ACM⁺22], ENT[RSM⁺22], DINOv2 [ODM⁺23], and DINOv3 [SVS⁺25] replaced or complemented centering and sharpening with Sinkhorn-Knopp normalization similarly to SwAV [CMM⁺20].

Contrastive Learning. Early modern SSL success came from contrastive learning methods [TKI20] that pull together augmented views of the same image (positive pairs) while repelling other images in the batch (negative pairs) in the embedding space, usually leveraging the InfoNCE [OLV18] objective for training. In this regard, MoCo [HFW⁺20b] narrowed the gap between the performance of self-supervised and supervised representation learning by introducing a momentum encoder and a queue to maintain a large pool of negatives, which turns out to be crucial for effective contrastive pretraining. SimCLR [CKNH20] simplified the recipe, emphasizing the usage of heavy augmentations and large batches. These works established the efficiency of contrastive pretraining and led to many successors that further improved the training recipe [CFGH20, CXH21b], scaled up pretraining [CKS⁺20], improved the utilization of positives [DAT⁺21, WWSY20, ZZL⁺21b] and negatives [HKW⁺22, CHT⁺21, WWW⁺21, YHH⁺22, ZZP⁺22, ITAC18, KSP⁺20, RCSJ20], explored the temperature parameter [KAG22, KBS⁺23], and learned dense features [WZS⁺21, HKA⁺21]. Beyond SSL, contrastive loss has been widely used in other fields, including multimodal

learning [RKH⁺21a, IWW⁺21, RBH⁺21, SCR⁺22], supervised learning [KTW⁺20], domain adaptation [TL21, HGX⁺22], domain generalization [YBZ⁺22, KYP⁺21], or metric learning [MMRM17].

Information Maximization and Teacher–Student Methods. This family of methods also relies on learning embeddings that are invariant to distortions of the input sample but, unlike contrastive approaches, avoids representational collapse by using information-maximization principles or teacher–student architectures without explicit negatives. In the line of information-maximization work, Barlow Twins [ZJM⁺21] and VICReg [BPL21] regularized cross-correlations and variance/covariance of twin embeddings reducing redundancy, and Whitening [ESSS21] constrained batch samples to lie on a spherical distribution. BYOL [GSA⁺20] avoided collapse using two networks, an online model (student) and an exponential moving average (EMA) target model (teacher), an asymmetric predictor applied only to the online model, as well as batch-normalization layers. SimSiam [CH21b] went further, showing that collapse can be prevented even without an EMA target model by using batch-normalization layers, an asymmetric predictor, and a stop-gradient on one branch. DINO [CTM⁺21], that was already mentioned in clustering-based approaches, further utilized a teacher–student setup but similarly to a knowledge-distillation idea [GYMT21]: instead of matching raw embeddings, the student matches the output soft distribution from the teacher. In practice, these methods reduce reliance on the large batches and explicit negatives needed in contrastive learning and work particularly well with ViT backbones [CTM⁺21]. However, they often require careful tuning of additional regularization mechanisms to avoid collapse [LEP22].

Masked Modeling. Masked modeling methods train representations by masking a portion of an input sample and learning to reconstruct the missing content. This approach is proven to be effective in language modeling, for example with masked autoencoding in BERT [DCLT19] and autoregressive modeling in GPT models [RWC⁺19]. Earlier vision work includes denoising autoencoders [VLBM08] with various image corruptions [VLL⁺10], including masking pixels [VLL⁺10, PKD⁺16, CRC⁺20]. However, unlike language, which provides information-dense discrete signals (e.g., words), images lack a discrete vocabulary and contain redundant information, which complicates training. Recent work addressed these issues in different ways. BEiT [BDPW21] replaced raw pixel regression with discrete visual tokens from a tokenizer to improve semantic supervision, and MaskFeat [WFX⁺22] predicted hand-crafted features such as HOG [DT05] instead of pixels. MAE [HCX⁺22] reconstructed pixels from highly masked inputs using a lightweight decoder and an asymmetric encoder–decoder design that makes pretraining efficient. SimMIM [XZC⁺22] showed that a simple recipe that regresses pixels with a small head is competitive at scale. These ideas were further extended in many follow-up works [CDW⁺24, DBZ⁺23, BHX⁺22, PDB⁺22, WBD⁺23, ADM⁺23]. Related work such as iBOT [ZWW⁺21] linked masked modeling and self-distillation by predicting masked token assignments in a teacher-student setup. In practice, masked image modeling provides strong pretraining for ViT architectures [DBK⁺20] and transfers especially well to dense prediction, although it often requires longer training than alternative objectives.

Generative Models. Recently, there has been growing interest in utilizing representations learned by generative models, namely diffusion [HJA20, SME20] and flow matching [LCBH⁺22] models, for various dense tasks. These models have shown significant advances in image generation [HJA20, SME20, LCBH⁺22], and it has further been shown that their pretrained backbones provide strong initialization for dense semantic tasks, including depth estimation [GAZS⁺25, HGL⁺25, KOH⁺24, FYH⁺25], surface normal estimation [GAZS⁺25, FYH⁺25], semantic segmentation [KA24], and object detection [CSSL23]. Recent works also explored how the representations learned by diffusion models can be adapted for discriminative objec-

tives [YW23], and how their features can be used in a training-free regime for zero-shot image classification [CJ23, LPD⁺23], image segmentation [HMH⁺25, NSF24, BRLR22], and semantic correspondence [TJW⁺23]. It has also been shown that features from generative models are complementary to those obtained from other self-supervised methods [ZHH⁺23], such as DINOv2 [ODM⁺23]. While this direction is very promising, training generative models is computationally expensive, and many design choices for extracting features for discriminative tasks remain open questions.

Video Representation Learning. Early video representation learning approaches leveraged pretext tasks such as frame ordering [MZH16a, MZH16b, LHSY17, FBGG17, XXZ⁺19], forward/backward playback discrimination [WLZF18], or playback rate classification [BEL⁺20]. Sermanet et al. [SLC⁺18] introduced contrastive training with a triplet loss that enforces a margin between positive and negative pairs, using frames from different viewpoints at the same timestamp as positives and frames from different timestamps as negatives. Han et al. [HXZ19, HXZ20] later leveraged the InfoNCE contrastive loss [OLV18] to predict future spatio-temporal features, using spatial and temporal features from other positions as negatives. Following the success of contrastive learning in images, it became the popular paradigm for video representation learning as well, where augmented clips of the same video, spatially or temporally overlapping segments, or spatial-temporal crops from a longer video were used as positive pairs, and clips from different videos were used as negatives [QMG⁺21, FFX⁺21, PHM⁺21, DGRS22, PSY⁺21, WBTT22]. Further, MAE-ST [FLH⁺22], VideoMAE [TSWW22], VideoMAEv2 [WHZ⁺23], and V-JEPA2 [ABF⁺25] demonstrated that masked reconstruction of spatio-temporal patches is also highly effective for scalable video pretraining. Some works [WWC⁺23, TLWB21] also combined masked modeling with contrastive learning objectives, enabling models to capture both fine-grained local structure and coarse-grained semantic similarity in video representations.

2.1.2 Multimodal Pretraining

In contrast to self-supervised learning, multimodal pretraining leverages co-occurring signals across modalities, such as images with captions or videos with synchronized audio, to learn robust representations. While earlier multimodal approaches relied on human-annotated datasets, e.g., COCO [LMB⁺14] with image-caption pairs, later works [RKH⁺21a, SBV⁺22, MZA⁺19, WLL⁺24] have significantly advanced the field by exploiting weakly aligned, web-scale supervision, e.g., LAION-5B [SBV⁺22] with 5B image-alt-text pairs and HowTo100M [MZA⁺19] with instructional videos and ASR-extracted narrations. In the following, we consider different approaches in multimodal pretraining.

Contrastive Pretraining. Contrastive learning has been especially influential in multimodal settings, aligning representations across paired modalities by bringing positive examples (e.g., an image and its corresponding caption) closer in representation space while discouraging similarity between negatives (e.g., an image and an unrelated caption), optimizing cross-modal retrieval. Earlier methods used canonical correlation analysis [HSST04, AABL13] to maximize correlation between projected vectors of images and text [RMV07, GKIL14, GWH⁺14] or margin-based ranking losses that penalize negatives ranked above the correct match [WBU11, FCS⁺13, KSZ14]. These methods were further extended by combining cross-modal learning with structure preservation within modality to stabilize alignment [WLL16], pairing cross-modal learning with caption generation [PMY⁺16, XXCC15] or with matching objectives [WLHL18], scaling to web-scale data with weak multimodal su-

pervision, e.g., leveraging noisy web tags alongside human-annotated captions [MPPRC18], extending to video–text learning [DLX⁺19, MLS18, MLMRC18], and enriching visual representations using mixtures of experts that integrate complementary cues such as faces, motion, appearance, and audio [MLS18, MLMRC18]. Further, ConVIRT [ZJM⁺22] proposed to utilize a transformer-based language encoder and train the model using InfoNCE contrastive objective [OLV18], showing excellent results for image–text pretraining in the medical domain. However, the success of contrastive pretraining with the InfoNCE objective was largely driven by scaling up training data using web sources, as demonstrated in CLIP [RKH⁺21b], which aligned images with their corresponding alt-text collected from the internet. Similarly, in the video domain, noisy but web-scale datasets have proven effective, such as HowTo100M [MZA⁺19] with instructional videos and narrations obtained via speech recognition, and WebVid-2M/10M [BNVZ21a] with videos and short descriptions from stock content platforms, that have enabled large-scale training of video–language models [MZA⁺19, MAS⁺20, BNVZ21a], as well as joint video–audio–language representations [RBH⁺21, AYQ⁺21]. Later works extended large-scale contrastive pretraining in various directions, including CLIP adaptations to video-text models [LJZ⁺22, MXS⁺22, XGH⁺21], architectural and loss changes [ZMKB23, ZWM⁺22, YHH⁺21, LLZ⁺21c], broader modality alignment, e.g. ImageBind for six modalities [GENL⁺23], as well as combination contrastive objectives with other losses in hybrid approaches, which we discuss later in this section. A large part of follow-up work also focused on scaling, cleaning, and open-sourcing pretraining data and frameworks, including ALIGN [JYX⁺21a], LAION-5B [SBV⁺22], OpenCLIP [IWW⁺21], and DataComp [GIF⁺23], InternVid [WHL⁺23], VAST [CLW⁺24], and EVA-CLIP [SFW⁺23, SWY⁺24].

Matching. Another widely used approach for multimodal pretraining is based on the matching objective [AZ17, OE18], which explicitly train a model to determine whether a given input pair (e.g., an image and a caption, or a video and an audio clip) corresponds to each other or not. The difference to contrastive pretraining lies in a mini-batch aggregation: contrastive methods compare positive examples against all negatives in a mini-batch, whereas matching treats each input pair independently and classifies it as either positive or negative. This idea was popularized in early audio–visual correspondence tasks [AZ17, OE18], where models learned to determine if a video segment matches an audio clip, providing a powerful self-supervised signal without annotations. Later, pixel- or region-wise matching tasks [GFG18, ZGR⁺18, GG19, ZGMT19, GHZ⁺20] were proposed for finer-grained alignment. A related concept was introduced in language modeling as “Next Sentence Prediction” objective in BERT [DCLT19], which classifies whether two sentences follow each other in text. Inspired by this, *Image–Text Matching* (ITM), also referred to as *Visual–Text Matching* (VTM), became a core pretraining objective for vision–language models such as LXMERT [TB19], VisualBERT [LYY⁺19], ViLBERT [LBPL19], UNITER [CLY⁺20], BLIP [LLXH22b, LLSH23], FLAVA [SHG⁺22], mPLUG [LXT⁺22], and others [KSK21, LSG⁺21a, LDF⁺20, HZL⁺20], where a binary classifier predicts whether an image–caption pair is aligned. The objective was later extended to the video–language setting, e.g., in HERO [LCC⁺20], mPLUG-2 [XYX⁺23], UMT [LWL⁺23], and the video–audio–language setting, e.g., in VAST [CLW⁺24]. In most frameworks, ITM is combined with other objectives, such as contrastive or masked prediction, where the contrastive loss also provides strong negatives for ITM training [LSG⁺21b, LLXH22b, LXT⁺22, XYX⁺23, CLW⁺24]. Unlike contrastive dual-encoder architectures, ITM matching is used in early/late fusion architectures, where tokens from different modalities attend to each other directly [LLXH22b, LXT⁺22, XYX⁺23, CLW⁺24]. A key strength of the matching objective compared to contrastive pretraining is its ability

to capture fine-grained cross-modal correspondences (e.g., local alignments between objects and words) that are not represented by global similarity scores, which is particularly useful for reasoning-based tasks such as visual question answering [CLY⁺20]. For this reason, ITM heads are also often used as re-ranking modules on top of contrastively trained dual encoders [LLXH22b, LWL⁺23, LXT⁺22, XYY⁺23].

Masked Pretraining. Masked objectives extend unimodal masked modeling [DCLT19, BDPW21] to the multimodal setting by predicting masked tokens or features in one modality conditioned on context from the same and other modalities [TB19, CLY⁺20]. Such masked prediction encourages fine-grained modality alignment, making it particularly effective for dense multimodal tasks such as visual question answering or referring expression comprehension [CLY⁺20]. However, in practice, similar to the matching objectives, masked modeling is usually combined with contrastive or matching losses in multimodal pretraining [TB19, LBPL19, CLY⁺20, LLXH22a]. Early vision-language models leveraged *Masked Language Modeling* (MLM) together with *Masked Region or Feature Modeling* (MRM), e.g., regressing or classifying masked visual regions while predicting masked words, typically using late-fusion architectures [TB19, LBPL19, CLY⁺20, LYY⁺19, HZL⁺20, LDF⁺20, ZZL21a, ZLH⁺21a]. Later models advanced this idea by jointly applying masked prediction to both image and text tokens, e.g., FLAVA [SHG⁺22], which combines BEiT-style image masking and BERT-style language masking into a *Masked Multimodal Modeling* objective. BEiT-3 [WBD⁺23] further unified masked modeling for vision and language within a single transformer trained jointly across modalities. In the video-language domain, joint masked modeling was introduced in VIOLET [FLG⁺21], which reconstructs masked text and spatio-temporal visual tokens. UniVL [LJS⁺20b] combined masked language modeling with masked frame modeling. However, later analysis [CWL⁺23] showed that while masked video modeling provides modest performance gains, it substantially increases computation. As a result, many recent video-text models, e.g., mPLUG-2 [XYY⁺23], X2-VLM [ZZL⁺23], All-in-one [WGY⁺23], and InternVideo2 [WLL⁺24], rely only on masked language modeling during pretraining. In the audio-visual domain, models such as AV-HuBERT [SHLM22], MAViL [HSX⁺23] and AV-MAE [GFI⁺23], as well as CAV-MAE [GRL⁺22] and CAV-MAE-Sync [ARG⁺25], perform joint masked modeling of both modalities, reconstructing masked audio and video tokens to learn robust cross-modal representations.

Hybrid Methods. Since many modern models employ multiple objectives and multiple architectural pathways, this paragraph summarizes hybrid approaches that combine contrastive alignment, matching, masked prediction, and sometimes generative losses, e.g., captioning, to capture complementary strengths. A prevalent recipe couples a *dual-encoder module* trained with contrastive loss for effective large-scale pretraining and high-level embeddings useful in retrieval and zero-shot transfer with a *fusion module* trained using matching, masked, or generative objectives for fine-grained grounding and reasoning. Hybrid vision-language models include ALBEF [LSG⁺21b], BLIP/BLIP-2 [LLXH22b, LLSH23], FLAVA [SHG⁺22], X-VLM [ZZL21a], and BEiT-3 [WBD⁺23], among many others [CLY⁺20, ZLH⁺21a, WGY⁺23]. Hybrid models are also popular in the video-text and video-audio-text domains, for example, HERO [LCC⁺20], UniVL [LJS⁺20b], and VIOLET [FLG⁺21] combine contrastive or matching losses with masked prediction and/or captioning. Recent video-text systems, e.g., UMT [LWL⁺23], mPLUG-2 [XYY⁺23], X2-VLM [ZZL⁺23], All-in-one [WGY⁺23], and InternVideo2 [WLL⁺24], retain contrastive pretraining for scalability while relying on ITM, MLM, or caption generation to improve localization and reasoning. Audio-visual models, e.g., AV-HuBERT [SHLM22], MAViL [HSX⁺23], AV-MAE [GFI⁺23], and CAV-MAE [GRL⁺22, ARG⁺25], analogously mix masked reconstruction with matching and/or contrastive terms to enforce cross-stream consistency. The video-audio-text model VAST [CLW⁺24] likewise unifies contrastive alignment

with matching and caption-generation objectives. In practice, modern pretraining strategies optimize a cocktail of losses, which leads to higher computational cost and increased sensitivity training curriculum and loss weighting, but the empirical gains have made hybrid training popular for large-scale multimodal pretraining.

2.1.3 Connection to Our Work

In this thesis, we make contributions to *contrastive learning* methods for self-supervised and multimodal pretraining.

In Chapter 3, we revisit the standard InfoNCE objective [OLV18] for contrastive training. In contrast to previous SSL contrastive methods that focus on refining the training recipe/scale [CFGH20, CKS⁺20] or exploring positives/negatives [DAT⁺21, RCSJ20] and temperature parameter [KBS⁺23], we focus on the objective itself. We propose a new objective that shifts optimization from *pairwise* similarities (as in InfoNCE) to local neighborhood structure, showing significant gains over InfoNCE in k -NN evaluations and linear probing, and competitive results against contemporary non-contrastive methods [SPK⁺23]. While masked and generative SSL methods have recently gained popularity, contrastive objectives remain a core tool that extends far beyond SSL [RKH⁺21a, TL21, MMRM17]. The proposed contrastive objective complements masked and generative SSL and can integrate into modern pretraining pipelines.

In Chapter 4, we focus on contrastive multimodal pretraining. Most contrastive approaches align modalities into a joint embedding space with dual encoders [RKH⁺21a, LHH22, AYQ⁺21], which works well for coarse-level tasks, e.g., retrieval, and leads to strong transfer performance, but does not allow to obtain *fused* representation of multiple modalities. For modality fusion, matching and masked objectives are commonly used, however, they enable learning token-level representations suited to dense tasks, e.g., VQA [LSG⁺21b, SHG⁺22, FLG⁺21]. In contrast, we introduce a model that fuses information from arbitrary modality combinations into a single joint embedding and train it with a *combinatorial* contrastive objective [SCR⁺22]. The resulting representation improves retrieval and zero-shot action localization by leveraging fused modalities (e.g., video-audio). Moreover, our pipeline remains fully contrastive, yielding a fused coarse-grained representation, which is complementary to matching/masked pretraining.

2.2 ADAPTING REPRESENTATIONS TO DOWNSTREAM TASKS

Representations learned from large-scale unlabeled web data through self-supervised or multimodal pretraining have proven effective across a broad spectrum of downstream applications, including image, video, and audio classification [CKNH20, TSWW22, RKH⁺21a, LJZ⁺22, AYQ⁺21, ARG⁺25], object detection and segmentation [CTM⁺21, HFW⁺20a], temporal action localization [RBH⁺21, MZA⁺19], cross-modal retrieval [RKH⁺21a, MZA⁺19, LLXH22a], VQA [SHG⁺22, LLXH22a], captioning [SHG⁺22, LLXH22a], and referring expression grounding [CLY⁺20, XYY⁺23]. Beyond generic vision domains, these representations have also been successfully applied in specialized settings, such as medical imaging [ZXQ⁺24, KARX24] and remote sensing [HCZ⁺25].

Adapting these strong general representations to downstream tasks is typically achieved through *zero-shot* transfer or *finetuning* (which can range from full model finetuning to parameter-efficient or few-shot strategies). Zero-shot transfer includes zero-shot “out-of-the-box” prediction, e.g., cross-modal reasoning tasks such as retrieval [RKH⁺21a, MZA⁺19], or zero-shot prompting, e.g., where handcrafted prompts (e.g., “an image of {class}”) map label

text into the shared multimodal embedding space to enable zero-shot classification, segmentation, or other tasks [RKH⁺21a, ZLD22, BPFK24]. Full finetuning using annotated downstream data generally achieves a great performance boost [LKS⁺23, SKSK23] but usually harms generalization [KRJ⁺22]. While parameter-efficient adaptation methods enable effective finetuning of large models when only limited downstream data is available and also better preserving generalization [HSW⁺21a, HWC⁺24, JTC⁺22, GGZ⁺24]. Such methods include prompt tuning methods, e.g., CoOp [ZYLL22b], CoCoOp [ZYLL22a], VPT [JTC⁺22], which learns textual or visual prompts while freezing the backbone, and adapters, e.g., AdaptFormer [CGT⁺22] and LoRA [HSW⁺21b], which insert small trainable modules that retain pretrained knowledge with only a fraction of trainable parameters. Linear probing with frozen backbone is usually used as a standard proxy to evaluate representation quality [CKNH20, HFW⁺20a]. However, finetuning strategies still require annotated datasets for downstream tasks and zero-shot strategies generally underperform finetuning approaches.

Since Part II, Finetuning Multimodal Models Without Human Supervision, focuses on adaptation strategies for action recognition and text–video retrieval, we discuss these tasks in more detail.

2.2.1 Action Recognition

SSL-pretrained video backbones, e.g., MAE-ST [FLH⁺22], VideoMAE [TSWW22] or VideoMAEv2 [WHZ⁺23], adapted for action recognition on benchmarks, e.g., Kinetics [KCS⁺17, CNB⁺18, CNHZ19] and SomethingSomethingV2 [GEKM⁺17a], through *supervised finetuning*, have established strong baselines. Foundation image–language models, e.g., CLIP [RKH⁺21a], have further enabled *zero-shot* action recognition by leveraging middle-frame embeddings, as demonstrated in CLIP [RKH⁺21a], or through temporal pooling of frame-level embeddings, as demonstrated by ViFi-CLIP [RKM⁺22]. However, as shown in Action-CLIP [WXL21], X-CLIP [NPC⁺22], ViFi-CLIP [RKM⁺22], and A5 [JHZ⁺22], finetuning image–language models on action recognition data, typically the Kinetics [KCS⁺17] dataset, and training temporal modeling modules remains crucial for achieving strong zero-shot and few-shot performance. Multimodal pretraining on large-scale video–text datasets, e.g., HowTo100M [MZA⁺19], also enables zero-shot action recognition, however, such models often underperform image-based variants partially due to the smaller scale and lower diversity of video–text data [MAS⁺20]. Recent works therefore combine both image– and video–text data during pretraining [LWL⁺23, LWL⁺23, WHL⁺23, WLL⁺24] to close the gap with finetuned action recognition. Nevertheless, even for large pretrained models, intermediate finetuning on curated action datasets, e.g., Kinetics, has been shown to provide additional performance gains [LWL⁺23, LWH⁺22].

2.2.2 Text–Video Retrieval

HowTo100M [MZA⁺19] introduced a large-scale narrated video dataset and demonstrated that video–text contrastive pretraining can transfer to text–video retrieval benchmarks such as MSR-VTT [XMYR16], YouCook2 [DXDC13], and LSMDC [RRTS15]. Using frozen visual backbones and learning projections into a shared video–text embedding space, the model already outperformed several methods trained directly on downstream datasets. However, finetuning on target domains provided substantial additional gains, highlighting the importance of domain-specific training. Follow-up work [MAS⁺20] advanced this by train-

ing the video encoder end-to-end on HowTo100M, further improving transfer across retrieval benchmarks. Frozen-in-Time [BNVZ21a] showed that pretraining on smaller but cleaner web video–text data leads to stronger retrieval performance compared to models pretrained on the noisier HowTo100M [MZA⁺19]. Foundation image–text models, e.g., CLIP [RKH⁺21a], have further boosted zero-shot text–video retrieval through strong visual–text alignment learned from large-scale and diverse web image–text data. To apply CLIP to video retrieval, CLIP4CLIP [LJZ⁺22] encoded individual frames and aggregated them temporally, achieving strong zero-shot transfer without explicit video pretraining, however, finetuning on noisy web video–text datasets, e.g., HowTo100M, led only to marginal improvements. Nevertheless, finetuning these strong pretrained models on downstream retrieval datasets, e.g., MSR-VTT [XMYR16], YouCook2 [DXDC13], or LSMDC [RRTS15], still results in significant performance gains [MAS⁺20, BNVZ21a, LJZ⁺22]. More recent works have pushed zero-shot retrieval performance further by improving web video–text dataset construction and annotation [WHL⁺23, CLW⁺24], combining image–text and video–text data [LWL⁺23, WHL⁺23, XYY⁺23], distilling knowledge from image models [LWL⁺23], and utilizing fusion models for re-ranking coarse retrievals [XYY⁺23, CLW⁺24]. Nevertheless, finetuning on downstream retrieval datasets continues to provide substantial performance boosts [CLW⁺24, LWL⁺23], emphasizing the importance of domain adaptation.

2.2.3 Connection to Our Work

In this thesis, we investigate finetuning strategies for adapting pretrained representations to specific tasks or domains without relying on annotated domain data.

In Chapter 5, we study adaptation of image–language models for action recognition. In contrast to prior work [WXL21, RKM⁺22, JHZ⁺22], which relies on supervised finetuning on action recognition datasets such as Kinetics [KCS⁺17], we show that effective finetuning can be achieved using *unlabeled* videos [LKS⁺23]. Moreover, by not restricting the textual space to classification vocabularies, our approach not only significantly boosts performance compared to zero-shot transfer but even surpasses some models trained in a fully supervised manner. The proposed strategy builds on strong image–text foundations and can be applied to newer image-text models such as EVA-CLIP-18B [SWY⁺24] and MetaCLIP2 [CLW⁺25].

In Chapter 6, we study the adaptation of vision–language models for text–video retrieval tasks. We show that it is possible to bridge the gap to domain-specific finetuning by relying only on *domain-specific language*, adapting the pretrained model to the text style of the target domain without using any paired video–text data or even target-domain videos [SKSK23]. In contrast to standard models that adapt through finetuning on curated video–text pairs [BNVZ21a, LJZ⁺22, CLW⁺24, LWL⁺23], our style-based transfer operates label-free and improves retrieval in the target domain.

2.3 IMPROVING DATA QUALITY

While recent advances in pretraining and adaptation strategies have substantially improved performance across various vision tasks [LLXH22a, LWL⁺23], the quality of training and evaluation data remains a crucial factor for developing robust models [GIF⁺23]. In this section, we review related efforts in the construction of large-scale web datasets without manual annotation and debiasing evaluation benchmarks to enable a more reliable performance assessment, as well as highlight connections to our contributions presented in Part III, Improving Data Quality

Without Human Supervision.

2.3.1 Improving Large-Scale Multimodal Datasets

Collecting large-scale web datasets inevitably introduces substantial noise, imbalance, and redundancy, making quality control an essential step in the creation of datasets. All large-scale dataset collection pipelines include steps to ensure data diversity and basic data quality. For example, HowTo100M [MZA⁺19] used WikiHow task descriptions [KW18] as search queries to diversify topics in the dataset, while CLIP [RKH⁺21a] employed 500,000 text queries and balanced the results by limiting the number of pairs per query. To ensure data quality, HowTo100M applied duration- and upload-based filtering and discarded videos with poor narration coverage, while CLIP filtered out non-English samples and removed corrupted files. Recent works demonstrate that more advanced data cleaning and refinement strategies are crucial for improving model performance [GIF⁺23, SVS⁺25, WHL⁺23], motivating new strategies to enhance data curation practices.

Filtering and Deduplication. Early datasets, such as Conceptual Captions [SDGS18], CC12M [CSDS21], and YFCC100M [TSF⁺16], implemented linguistic heuristics and content filtering to discard low-quality or offensive samples. Video-text datasets such as YouTube-8M [AEHKL⁺16], YT-Temporal-180M [ZLH⁺21a], HowTo100M [MZA⁺19], WebVid [BNVZ21a], WTS-70M [SLS⁺20], and VidChapters-7M [YNL⁺24] also incorporated basic quality control through duration, resolution, and language filtering, as well as duplicate removal based on video IDs or URLs. LAION-5B [SBV⁺22] applied CLIP-based similarity scoring and aesthetic filtering to retain semantically consistent and visually high-quality image-text pairs. BLIP [LLXH22a] used an image-text matching head finetuned on a clean, human-annotated dataset to filter noisy image-text pairs based on alignment confidence. Subsequent works, such as DataComp [GIF⁺23], MetaCLIP [XXT⁺23], DINOv2 [ODM⁺23], and DINOv3 [SVS⁺25] further improved a data curation pipeline, introducing large-scale deduplication techniques, text-based filtering, and resolution and aspect-ratio constraints to remove uninformative or low-quality samples and maintain data diversity.

Temporal Alignment. Instructional videos with textual narrations obtained from automatic speech recognition (ASR) provide a rich source of long and densely annotated video-audio-text data, including such datasets as HowTo100M [MZA⁺19], YT-Temporal-180M [ZLH⁺21a], and HD-VILA-100M [XHZ⁺22]. However, the problem of noisy and temporally misaligned ASR narrations in such datasets has been widely recognized and has been addressed in several works [MAS⁺20, ABARB21]. Since ASR segmentations often cut sentences mid-phrase, adding noise to the temporal alignment, Zellers et al. [ZLH⁺21a] improved transcripts by restoring punctuation and, therefore, sentence boundaries, correcting transcription errors, and segmenting long narrations, and then to improve temporal alignment of the new segmentation, performed Dynamic Time Warping [Mulo7] based on word-level similarity to the original ASRs. Han et al. [HXZ22] further proposed alignment networks to automatically filter out non-alignable narrations and refine temporal alignment for the others. Some works did not correct temporal alignment directly but instead proposed loss functions to handle misaligned supervision, e.g., MIL-NCE [MAS⁺20] and soft max-margin ranking loss [ABARB21].

Improving Textual Annotation. Since textual data collected from the web often contains substantial noise, such as grammatical errors, incomplete phrases, and irrelevant or poorly written sentences, various works have focused on improving the quality of text supervision. Zellers et al. [ZLH⁺21a] improved textual narrations of instructional videos through lightweight correc-

tions, such as punctuation restoration and transcription error fixes. Lin et al. [LPB⁺22] matched noisy narrations to WikiHow step descriptions [KW18]. BLIP [LLXH22a] trained a multimodal model with multiple objectives, including image caption generation, on a diverse mixture of datasets combining large-scale web and curated data, and then finetuned the captioning head on curated data to generate new captions for web data, effectively improving textual supervision. Follow-up works such as Recap-DataComp-1B [LTH⁺24] and DreamLIP [ZZW⁺24] further regenerated captions using pretrained vision-language models, while LaCLIP [FKI⁺23] rewrote captions using LLM. Recent video datasets, such as InternVid [WHL⁺23] and Panda-70M [CSM⁺24], Video-ChatGPT [MRKK23], extended this idea to the video domain by generating large-scale textual narrations via per-frame captioning and LLM-based summarization, while HowToStep [LCH⁺23] summarized ASR transcripts into procedural steps using LLMs. Instead of relying solely on pretrained captioning models, VeC [LZZ⁺24] aggregated information from alt-text and generated captions, while VAST [CLW⁺24] integrated ASR transcripts, audio captions, and video descriptions to create textual descriptions of video clips.

2.3.2 Debiasing Evaluation Benchmarks

Most visual datasets are prone to various forms of bias, which have been systematically analyzed in surveys [TE11, FPNK22, MMS⁺21]. In this section, we focus specifically on *representation bias*, where predictions may rely on correlated or contextually related cues (e.g., scene background) instead of reasoning about the visual content genuinely required to recognize the intended action or concept. In this regard, a unique effect in video understanding is the strong impact of object and scene information on the overall performance. Different strategies have so far been explored to mitigate this bias. He et al. [HSSK16] and HAT [CWR22] proposed to separate the performance of scene appearance from human motion by segmenting human figures and placing them into black/gray background, or alternatively by cropping or inpainting human figures and classifying videos based on background only. Weinzaepfel et al. [WR21] proposed the Mimetics dataset by extending certain classes from the Kinetics dataset [KCS⁺17] with videos of activities in unrelated background. Zhai et al. [ZLW⁺23] proposed an approach for scene-debiased action recognition via adversarial scene reconstruction and classification. Another line of work focused on capturing datasets with explicitly low object and scene bias. SomethingSomethingv1 and later v2 dataset [GEKM⁺17b] were proposed as an almost scene-free dataset, focusing on simple manipulative actions only. Li et al. [LLV18] proposed a procedure for designing representation unbiased datasets and introduced Diving48 as an example. However, this specialization usually also implies that those cases are rather suitable for specialized approaches but not for, e.g., zero-shot evaluation of generalist video models. Other studies [BEG⁺22, LBB23] showed that for several benchmarks, using single frames during training or inference can achieve performance comparable to models that incorporate temporal information. However, recent efforts have mainly focused on developing new benchmarks to assess temporal understanding in videos [BTS23], e.g., visual question-answering benchmarks [MAM23, CTZ⁺24], while temporal representation bias in standard benchmarks has not been thoroughly investigated.

2.3.3 Connection to Our Work

In this thesis, we make contribution to improving both training data and debiasing evaluation benchmarks, focusing on video understanding.

In Chapter 7, we propose a method to improve textual annotations for instructional videos by transforming noisy ASR transcripts into human-written-like captions using LLMs, while also refining their temporal alignment and performing filtering [SKH⁺24]. In contrast to prior works [WHL⁺23, CSM⁺24, MRKK23] that generate new video descriptions through image captioning and summarization, namely transferring knowledge from image models to videos, our approach leverages existing noisy ASR supervision to uncover new knowledge within it. Unlike VAST method [CLW⁺24], which merges ASR narration, image caption, and audio caption into the final caption, we reason over long ASR context beyond single ASR segments to improve coherence. The most related work, HowToStep [LCH⁺23], which was concurrent to ours, focuses on generating procedural steps, while our method produces general-purpose video captions.

In Chapter 8, we propose a method to analyze and debias video benchmarks with respect to different types of undesired representation bias. In contrast to prior work [HSSK16, CWR22, LLV18, GEKM⁺17b, BEG⁺22, LBB23], our approach enables a systematic analysis of such biases in *existing* benchmarks *across multiple dimensions*, including various conceptual and temporal bias. Furthermore, to the best of our knowledge, we are the first to propose mitigating object representation bias in benchmarks by discarding object-biased samples, which allows us to avoid adding artifacts to test videos, as done through cropping or inpainting in [HSSK16, CWR22], and to keep the dataset domain unconstrained, unlike [GEKM⁺17b, LLV18].

2.4 GROUNDING WITH MULTIMODAL MODELS

Visual grounding aims to find the specific region in an image or video that corresponds to a given textual description, thus allowing not only to identify or classify a sample correctly, as in retrieval or classification tasks, but also to localize the evidence within it. Video grounding introduces additional challenges compared to image grounding, as it requires reasoning over the temporal dimension and understanding events across time, while also having the higher cost and complexity of collecting and annotating video data. As in Part IV, Grounding with Multimodal Models Without Human Supervision, we focus specifically on grounding in videos, this section discusses related work on spatial, temporal, and spatio-temporal video grounding.

2.4.1 Spatial Video Grounding

Spatial video grounding aims to localize the region in a video frame corresponding to a given natural language query. In contrast to temporal grounding, it is assumed that the referred concept is visible in the frame. Most existing works focus on localizing objects in videos based on a language query, a task known as *Referring Video Object Segmentation* (RVOS) [SLH20]. Gavriluk et al. [GGLS18] and Khoreva et al. [KRS18] were the first to move from localization of predefined classes in frames to the localization of actors and objects described by natural language sentences, introducing the A2D-Sentences, JHMDB-Sentences, and Ref-DAVIS17 datasets. Ref-YouTube-VOS [SLH20] and MeViS [DLH⁺23] further extended RVOS to longer and more diverse videos. However, RVOS methods, including earlier approaches [GGLS18, WDMY20, MDRS20, WDT19, NXWT20, SLH20, LHH⁺21], transformer-based models [BZB22, WJS⁺22, WWZ⁺23b, TZY23, HD24], and recent open-vocabulary methods [LLT⁺25, KJL⁺24], typically rely on supervised training with annotated RVOS datasets to achieve object grounding. Another line of work [WGX21, ZLC18a, SXGX19] considers weakly supervised object grounding in video, training on curated video-text pairs and object proposals but without ground-truth

bounding-box annotations. In contrast, Tan et al. [TPS⁺21] stand out as the first to focus not on localizing individual objects or actors, but localizing *interactions* that may involve multiple entities and actions and often lack clear spatial boundaries. To evaluate this setting, Tan et al. introduced the YouCook2-Interactions dataset, which provides bounding-box annotations for interactions described by natural language sentences. Moreover, Tan et al. proposed to train a spatial grounding model using only web video-text data from HowTo100M [MZA⁺19] without any spatial annotations, introducing Contrastive Multilayer Multimodal Attention (CoMMA) to learn cross-modal localization in a self-supervised manner via attention rollout.

2.4.2 Temporal Video Grounding

Temporal video grounding, also known as natural language moment retrieval [AHWS⁺17, ZDW⁺19], or temporal sentence grounding in videos [ZSJZ23] aims to localize the temporal segment in an untrimmed video that corresponds to a given language query. In contrast to spatial grounding, which focuses on identifying regions within a frame, temporal grounding determines when an event described in text occurs within the video. Early works [AHWS⁺17, GSYN17, LWN⁺18, CCM⁺18, ZPFL20, ZSJZ20, MCH20, LYBB20] approached this task under full supervision, requiring manually annotated start-end timestamps for each query, e.g., leveraging such datasets as DiDeMo [AHWS⁺17], CharadesSTA [GSYN17], TACoS [RRW⁺13], ActivityNet Captions [KHR⁺17], and TVR [LYBB20]. Later works [MPRC19, LZZ⁺20, TXSP21, GDSX19] aimed to reduce annotation costs by exploring weakly supervised settings, where paired video-text data are available but without precise temporal boundaries for each activity. Zhukov et al. [ZAC⁺19b] further proposed to use web supervision from narrations of instructional videos to temporally localize procedural steps, leveraging additional information about step order from WikiHow [KW18] and releasing the CrossTask benchmark for evaluation. Later video-text models [MZA⁺19, MAS⁺20, XGH⁺21, CRD⁺21, ZY20, LJS⁺20b, YMH⁺22] demonstrated that large-scale multimodal pretraining on web datasets such as HowTo100M [MZA⁺19] enables emergent zero-shot temporal localization abilities. Transformer-based approaches, e.g., Moment-DETR [LBB21], QD-DETR [MHP⁺23], UniVTG [LZC⁺23], and LD-DETR [ZHZ⁺25] further showed that specialized architectures for moment retrieval benefit from large-scale web data, either by leveraging strong pretrained embeddings (e.g., CLIP [RKH⁺21b]) or by pretraining their own models on web video-text corpora. Recent multimodal language models, such as VideoChat-TPO [YLH⁺25] and InternVideo-2.5 [WLY⁺25], include temporal grounding as one of many pretraining tasks based on supervised datasets, thereby enabling temporal grounding abilities in large multimodal language models.

2.4.3 Spatio-Temporal Video Grounding

Spatio-temporal video grounding aims to localize where and when an event described in natural language occurs in an untrimmed video, combining the goals of both spatial and temporal grounding. Early research on spatio-temporal localization focused purely on detecting fixed number of categories of human actions with explicit bounding-box and timestamp supervision, as in AVA [GSR⁺18], DALY [WMS16], and UCF101-24 [SZS12]. The HC-STVG [TLL⁺21] and VidSTG [ZZZ⁺20] datasets marked a key step toward natural language-conditioned spatio-temporal grounding, introducing benchmarks for jointly localizing spatio-temporal tubes around objects and humans based on natural language queries in uncurated videos.

Building on these datasets, several transformer- and graph-based models were proposed, including STGRN [ZZZ⁺20], STVGBert [SYX21], and STGVT [TLL⁺21]. Later methods, such as STCAT [JyLYM22], TubeDETR [YMS⁺22], and VideoGrounding-DINO [WNK⁺24], adopted DETR-style [CMS⁺20] transformer architectures, significantly improving cross-modal reasoning and enabling end-to-end grounding. More recently, Ji et al. [JLS⁺24] proposed a new benchmark, DVD-ST, where none or several objects in a video may correspond to the query, and introduced methods to adapt existing models to this more challenging setting. Similar to spatial and temporal grounding, several works [CBK20, LWZ⁺23, JM24, KKR25] explored weakly-supervised spatio-temporal grounding, training models without dense spatial or temporal annotations but relying on annotated video–text correspondences. Zhao et al. [ZYW⁺24] further applied visual prompting of the CLIP model [RNSS18] for spatio-temporal grounding, training an additional interaction module to filter out irrelevant frames. However, all the discussed datasets and methods share several limitations. Temporal detection is usually restricted to identifying only the start and end frames of a single action per video, while more complex and realistic scenarios, where longer videos comprise multiple sequential actions, addressed in some temporal localization datasets such as Cross-Task [ZAC⁺19b], remain underexplored. Moreover, existing datasets primarily focus on grounding objects and people, leaving activity grounding largely unexplored.

2.4.4 Connection to Our Work

In this thesis, we further advance spatial and spatio-temporal video grounding.

In Chapter 9, we explore spatio-temporal activity grounding in untrimmed multi-action videos [CSR⁺24]. In contrast to prior works [TLL⁺21, ZZZ⁺20, WNK⁺24, JM24] in spatio-temporal grounding that focus on grounding objects or actors, we address *action* grounding, aiming to find regions where actions occur, which often lack clear spatial boundaries and may involve multiple objects or actors. Moreover, our setting considers grounding in multi-action videos, unlike previous spatio-temporal grounding benchmarks [WMS16, TLL⁺21, ZZZ⁺20, JLS⁺24] that focus on single-action clips. To this end, we propose a new benchmark dataset, GroundingYouTube, the first with dense spatio-temporal annotations centered around activities in long, untrimmed instructional videos containing multiple actions. Furthermore, unlike prior works [JyLYM22, YMS⁺22, JM24] that rely on full or weak supervision, we propose a method that enables grounding using only multimodal supervision from web-narrated instructional videos, e.g., HowTo100M [MZA⁺19], without any human annotations. The closest to our work is the CoMMA method [TPS⁺21], which also introduced the YouCook2-Interactions dataset with bounding boxes for interactions, however, CoMMA focuses only on spatial grounding, while we address spatio-temporal grounding.

In Chapter 10, we further explore spatial action grounding. We propose a method that enables spatial grounding in pretrained multimodal models in a fully training-free manner [VBK⁺25]. To achieve this, we extend the GEM framework [BPFK24], originally developed for grounding objects and referring expressions in images, to the task of activity grounding in videos. In contrast to prior works [WWZ⁺23b, TZY23, HD24, LLT⁺25, KJL⁺24] that rely on full supervision, weakly supervised approaches [WGX21, ZLC18a, SXGX19], and the CoMMA method [TPS⁺21], which leverages web data but still requires training a specifically designed architecture for action grounding, our approach requires no training at all, enabling spatial grounding directly in contrastively pretrained video–language models.

I

PRETRAINING REPRESENTATIONS WITHOUT HUMAN SUPERVISION

In the first part of the thesis, we study learning generalizable representations without human annotations through multimodal and self-supervised pretraining. Our goal is to obtain robust representations with strong transfer to downstream image and video recognition tasks.

In Chapter 3, we propose a group-ordering-based contrastive loss that relaxes the standard contrastive objective commonly used in self-supervised and multimodal learning. Using this objective, we train image representations from raw images without labels and show that it produces more coherent local neighborhoods in embedding space and significantly improves downstream performance.

In Chapter 4, we explore learning from web multimodal video-audio-text data and focus on representations that encode not only single modalities (e.g., video-only or audio-only) but also fused representations for any combination of modalities. To this end, we propose a new architecture and loss function that learn to fuse multimodal features, improving cross-modal retrieval action localization using fused modalities.

LEARNING BY SORTING: SELF-SUPERVISED LEARNING WITH GROUP ORDERING CONSTRAINTS

Contents

3.1	Introduction	30
3.2	Related Work	31
3.2.1	Self-supervised Representation Learning	31
3.2.2	Differentiable Sorting and Ranking	33
3.3	Method	33
3.3.1	Training Pipeline	33
3.3.2	Group Ordering Constraints (GroCo)	34
3.3.3	Learning by Sorting	34
3.4	Experimental Evaluation	37
3.4.1	Implementation Details	37
3.4.2	Evaluation Procedure	38
3.4.3	Comparison to State-of-the-Art	38
3.4.4	Comparison to the Pairwise Contrastive Loss	39
3.4.5	Ablation Study	41
3.5	Conclusion	42

Contrastive learning has become a prominent ingredient in learning representations from unlabeled data: models are trained to pull together positive samples (e.g., augmented views of the same image) and push apart negative samples (e.g., different images) in embedding space. It is effective not only for unimodal representation learning, e.g., image representation learning from unlabeled images, but also as the backbone of multimodal alignment across images, video, audio, and text (e.g., pulling together a video and its caption). Despite this success, the standard contrastive objective has limitations. First, it reasons primarily over pairwise relations (e.g., pairs of positives and pairs of negatives), overlooking the structure of local neighborhoods. Second, it minimizes distances between all positives and maximizes separation from all negatives, which can be suboptimal: positives that share a concept may still differ (pose, viewpoint, context) and need not be collapsed, while far or easy negatives contribute little yet still influence the loss. This chapter proposes a new approach towards self-supervised contrastive learning based on Group Ordering Constraints (GroCo). The GroCo loss leverages the idea of comparing groups of positive and negative samples instead of pairs of samples. Building on the recent success of differentiable sorting algorithms, group ordering constraints enforce that the distances of all positive samples (a positive group) are smaller than the distances of all negative samples (a negative group); thus, enforcing positive samples to gather around an anchor. This leads to a more holistic optimization of the local neighborhoods. Using this approach, we train an image representation from unlabeled images and evaluate it on a suite of competitive self-supervised learning benchmarks. We show that our method is not only competitive to current methods in the case of linear probing but also leads to higher consistency in local representations, as can be seen from a significantly improved k -NN performance across all benchmarks.

This chapter is based on the ICCV 2023 paper [SPK⁺23]. As the first author, Nina Shvetsova contributed to the idea, designed the study and evaluation protocol, implemented the code, conducted all experiments and analyses, and contributed to the writing.

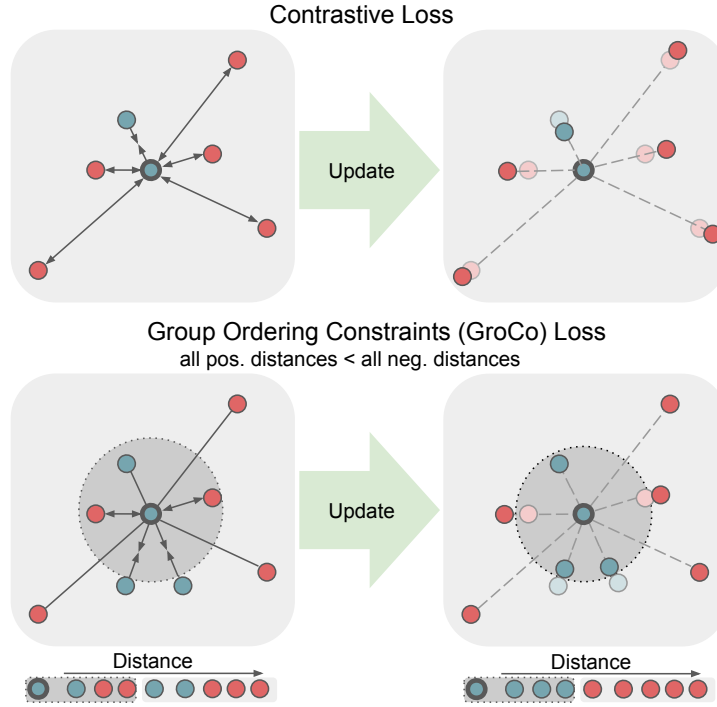


Figure 3.1: Pairwise contrastive loss compared to the proposed group ordering constraints loss: While pairwise loss only considers pairwise distances, groupwise sorting allows us to enforce that a group of positive samples is closer to the anchor than a group of negative ones and, thus, to improve the representation of the local neighborhood.

3.1 INTRODUCTION

Self-supervised learning has become a topic of growing interest over the last years as it allows models to learn representations from large-scale data without any need for annotation. Recently, self-supervised contrastive methods [DAT⁺21, CKNH20, TSP⁺20, HFW⁺20a, BHB19, YZYC19] notably narrowed the gap to the supervised learning performance. These methods rely on the concept of the pairwise contrastive loss, which is based on the idea that an image, serving as anchor, and an augmentation of the same image, a so-called positive pair, should be close to each other in a projection space, while a pair made up of an anchor image and a different image, a so-called negative pair, should be far away from each other. This concept of bringing positive pairs in embedding space closer together was further extended in various frameworks, such as BYOL [GSA⁺20], SwAV [CMM⁺20], and many more [ZJM⁺21, BPL21, ESSS21, CTM⁺21, CH21b].

In this work, we propose a novel approach to address those points by shifting away from the concept of a pairwise contrastive loss and instead propose to train the network based on Group Ordering Constraints (GroCo)—namely, the idea that the group of multiple positives should be closer to an anchor image than any negative from the group of negatives. We illustrate this idea and comparison with the pairwise contrastive loss in Figure 3.1. With group constraints, we do not treat positive and negative pairs individually, as in contrastive loss, but rather as groups, and effectively constrain only those elements that are placed in the group overlap area with respect to the distance to the anchor point.

To enforce the group ordering constraints in the projection space, we propose the idea

of *learning by sorting*: we suggest sorting positives and negatives by distance to the anchor image in a differentiable way and swapping them if they are in the wrong order. To create an end-to-end training pipeline, we leverage recent advances in differentiable sorting [PBKD21, GWZE19, CTV19, PBKD22]. Specifically, we utilize a differentiable sorting algorithm to obtain a differentiable permutation matrix for sorting a list of distances to the positive and negative images, as shown in Figure 3.2. If we would know the full ground truth orderings among positives and negatives, such as which positive sample should be closer to the anchor than another positive sample, we could create a ground truth permutation matrix, and calculate how much the predicted permutation matrix would deviate from the ground truth one [GWZE19, CTV19, PBKD21, PBKD22]. Since this assumption is not given in practice and we do not know the ground truth distance ordering within the positive or the negative groups, we propose the GroCo loss, a relaxed formulation of the original sorting supervision that captures how many negative elements appear in the positive positions and vice versa. Compared to the pairwise contrastive loss, the resulting group ordering focuses on *optimizing the local neighborhood* around an anchor image, rather than optimizing all data points at once. Moreover, without any additional modifications, our group ordering loss directly focuses on the strongest positive and negative examples.

To show the capabilities of the proposed approach, we evaluate it on a suite of competitive self-supervised learning benchmarks, namely in the context of linear probing, k -NN classification, as well as image retrieval. Our evaluation demonstrates that models trained via group ordering constraints obtain competitive performance in linear evaluation compared to other contrastive learning frameworks, some of them even trained with significantly larger batch sizes and/or an additional teacher network. Further, we demonstrate the superior ability of the proposed method in the context of shaping local neighborhoods by evaluating the nearest neighbor classification capabilities by outperforming any other state-of-the-art method on this task.

We summarize the contributions of this work as follows.

- We propose a new concept for self-supervised representation learning based on learning by sorting.
- We leverage recently proposed differentiable sorting methods to derive the contrastive group ordering constraints (GroCo) loss.
- We demonstrate that the proposed method achieves competitive performance in linear separability of embeddings and is especially suitable to model the local neighborhoods and outperforms current top-level state-of-the-art frameworks on a wide range of nearest-neighbor tasks.

3.2 RELATED WORK

3.2.1 Self-supervised Representation Learning

Self-supervised learning aims to learn a robust representation from data without human annotation. Various learning objectives were developed for self-supervised learning from images including image colorization [ZIE16], inpainting [PKD⁺16], puzzle solving [NF16], and rotation prediction [GSK18].

Over the last years, methods [CKNH20, HFW⁺20a, GSA⁺20, CMM⁺20, CH21b] that enforce the model to be robust to different image distortions achieved great performance improvements

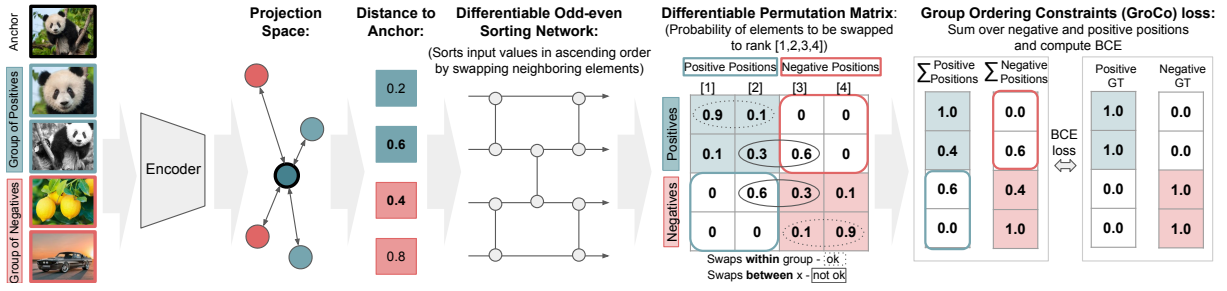


Figure 3.2: Overview of the proposed group ordering constraints loss: We first compute the distance of groups of positive as well as negative samples with respect to an anchor. The distances are sorted via a differentiable odd-even-sorting network in non-descending order, computing the *swapping probability for samples*. The result is a differentiable permutation matrix, in which the row values can be thought of as the probabilities of sorting the elements into the corresponding positions. We can ignore swap operations within groups, and only need to consider swapping operations between groups. We, therefore, sum over the positive and negative columns of the permutation matrix and compute the BCE between the row-wise entries and the expected ground truth.

in self-supervised learning. Such methods generally rely on sampling two augmented views of the image—a positive pair—and minimize the distance between those in the embedding space. Contrastive methods [CKNH20, HFW⁺20a, CKS⁺20, CFGH20] are a great example of those approaches. To prevent the model from learning a trivial solution for any input, contrastive methods also introduce the concept of a negative pair, i.e., two different image, and *contrast* positive pairs against negative pairs. The InfoNCE loss, which is often referred to as a contrastive loss, is the most prominent method in many self-supervised learning scenarios [CKNH20, HFW⁺20a, RKH⁺21b, AYQ⁺21]. Earlier contrastive methods relied on the triplet loss [SKP15], which considers one positive and one negative example for the anchor image. In contrastive learning, many components and extensions have been investigated: data augmentation strategies [CKNH20, XWED20], projection head design [CKS⁺20], hard negative sampling [RCSJ20], increasing the richness of positives with nearest neighbours [DAT⁺21], or mitigating effect of false negatives [HKW⁺22]. In this work, we propose a novel contrastive method—learning by sorting—which features properties not inherent to other contrastive learning methods: we primarily consider those positives and negatives that are placed (sorted) incorrectly in the embedding space, thereby implicitly considering the strongest positive and negative examples.

There are also methods [CH21b, GSA⁺20] that do not rely on negatives and only maximize agreement between positive views. Such methods prevent collapsing of the representation space by using asymmetric architectures applied to different views [CH21b, GSA⁺20], an additional teacher network [GSA⁺20, CTM⁺21], stop gradient [CH21b, GSA⁺20, CTM⁺21], feature whitening [ESSS21], or information maximization [ZJM⁺21, BPL21]. Another set of methods [CMM⁺20, CBJD18, ARV20] utilizes clustering of the latent embeddings. However, so far, methods mostly rely on similarity maximization between only two sampled views. SwAV [CMM⁺20] also considered sampling more augmentations in a multi-crop setting, where two full-size augmented images are sampled together with several smaller crops, and Whitening [ESSS21] utilized more full-resolution samples. However, even in these cases, the loss was still computed by considering pairs of views independently. In contrast, we propose optimizing the embedding space based on *group ordering constraints*.

3.2.2 Differentiable Sorting and Ranking

Differentiable sorting and ranking methods provide a pipeline that allows training neural networks with ordering supervision in an end-to-end fashion with gradient descent [GWZE19, BTBD20, PE20, CTV19, PBKD21, PBKD22]. The sorting operator can be seen as a function returning a permutation matrix that indicates the permutation necessary to sort the sequence of values (the matrix that multiplied with a input vector returns a sorted output vector.) In this context, differentiable sorting refers to relaxing the (hard) permutation matrix to a differentiable permutation matrix via continuous relaxations. The differentiable permutation matrix for a given sequence of values, which could, e.g., be scores predicted by a neural network, can then be used to compute the loss by comparison to a ground truth permutation matrix. Recently, multiple methods for relaxing the permutation matrix have been proposed, including an argsort approximation by unimodal row-stochastic matrices [GWZE19, PE20], a formulation of entropy-regularized optimal transport [CTV19], as well as networks of differentiable swap operations (differentiable sorting networks) [PBKD21, PBKD22]. The latter method composes the full permutation matrix as a product of permutation matrices that arise from comparing only two elements at a time (usually neighbors) and either swapping them or not swapping them. Inspired by the idea of swapping the neighboring elements in the sequence, our approach learns representations by comparing neighboring elements in the embedding space.

In terms of applications, differentiable sorting has been leveraged in various contexts, including recommender systems [LC]⁺21], image patch selection [CMD]⁺21], selection experts in multi-task learning [HZC]⁺21], and attention mechanisms [ZYW]⁺21]. To the best of our knowledge, the proposed method is the first work to leverage ordering supervision for self-supervised learning of visual representations.

3.3 METHOD

Given a dataset of images $\{x_i\}_{i=1}^M \subseteq \mathcal{X}$, our goal is to learn an encoder $g : \mathcal{X} \rightarrow \mathbb{R}^d$ that extracts image representations that later might be used for downstream tasks.

3.3.1 Training Pipeline

Similar to other contrastive methods, our approach considers several augmented views of the same image as positive examples, which should be close together in an embedding space, and different images as negative examples, which should be apart in an embedding space. Our model starts from sampling mini-batches of B images and generates $m \geq 2$ randomly augmented views per each image, resulting in $m \cdot B$ data points. If $m = 2$ we are close to the original pairwise contrastive learning setup [HKW]⁺22, CKNH20, CH21b, BPL21, ZJM]⁺21]. The augmented views are processed with the encoder network $g(\cdot)$, that extracts data representation vectors from views. Then, an MLP projection head $h(\cdot)$ maps representations to the latent space where distances between views are calculated. For each data point serving as an anchor x^a , we have $m - 1$ positive examples $\{x_i^p\}_{i=1}^{m-1}$ (which results in 1 positive example in the classical setup of $m = 2$) and $m \cdot (B - 1)$ negative examples $\{x_i^n\}_{i=1}^{m \cdot (B-1)}$. We use the cosine distance to measure the distance between data points: $d(x, y) = -\frac{x^\top y}{\|x\| \|y\|}$.

3.3.2 Group Ordering Constraints (GroCo)

In order to consider not only pairs of views but instead groups of multiple positives at once, we extend the contrastive loss to the idea that *the group of positives* should be closer to the anchor image than *the group of negatives* in the embedding space. This idea directly results in *group ordering constraints* (GroCo). To simplify the notation, we denote the distance between data point x^a and its positive x_i^p and negative examples x_i^n as $d_i^p = d(x^a, x_i^p)$ and $d_i^n = d(x^a, x_i^n)$. If we assume that K positives $x_1^p, \dots, x_K^p$ are ordered with respect to their distances to the anchor x^a as $d_1^p \leq \dots \leq d_K^p$ and N negatives $x_1^n, \dots, x_N^n$ as $d_1^n \leq \dots \leq d_N^n$, then we define the *group ordering constraints* as

$$d_1^p \leq \dots \leq d_K^p < d_1^n \leq \dots \leq d_N^n. \quad (3.1)$$

As can be seen in Equation 3.1, all elements in the groups are considered in the constraints; however, for training, the relevant constraint is the (bold) $<$ in the center. This constraint differs from pairwise contrastive loss constraints, which 1) consider positives individually and 2) consider all negatives even if they are far away from the anchor and are thus less relevant. Our constraints also differ from the triplet loss, which considers only one positive and one negative example, while we are working with groups.

3.3.3 Learning by Sorting

Inspired by recent advances in differentiable sorting [PBKD21, PBKD22], we design a loss that fulfills our group ordering constraints based on differentiable sorting. Our training procedure can be seen as sorting positives and negatives in the embedding space with respect to an anchor image and swapping them if they are in the incorrect order; therefore, we call the approach *learning by sorting*.

3.3.3.1 Differentiable Sorting Networks

In the following, we review the differentiable sorting algorithm called *differentiable sorting networks* [PBKD21] as it is a core element of our loss function. We note that “networks” in “sorting networks” is not related to the term “neural networks” and that a differentiable sorting network is a function with no trainable parameters. Sorting networks are a family of sorting algorithms from classic computer science literature [Knu98]. An example of this is the odd-even sorting network. By relaxing the conditional swap operator to a differentiable conditional swap, sorting networks can be made differentiable by relaxation to differentiable sorting networks [PBKD21].

We consider differentiable sorting networks based on the odd-even sorting network to sort an input sequence of $K + N$ elements in non-descending order as shown in Figure 3.3. The differentiable sorting network is defined as a composition of functions, where each function refers to one step of sorting, and where pairs of elements of the input sequence are compared and swapped if they are in the wrong order via a *conditional swap operation*. For the odd-even sorting network, the algorithm compares neighbored elements on odd and even indices alternatingly in each step, and requires $K + N$ steps to sort a given input sequence of length $K + N$.

The conditional swap operation for elements (d_i, d_j) where $i < j$ can be defined as $d_i' = \min(d_i, d_j)$, $d_j' = \max(d_i, d_j)$, and the differentiable relaxation [PBKD22] of this operation is

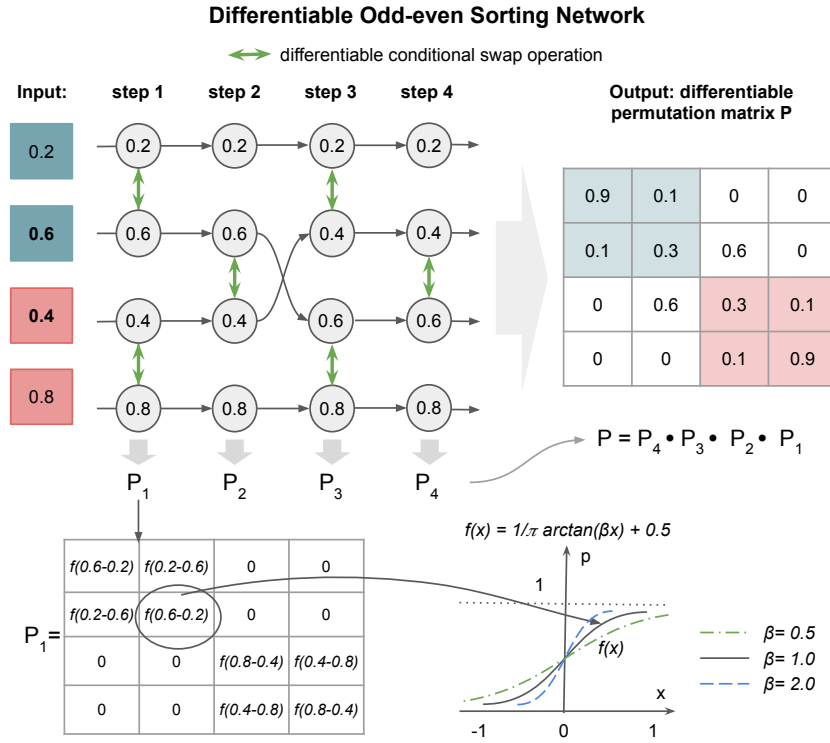


Figure 3.3: Overview of a differentiable sorting network with the odd-even sorting algorithm. The network compares neighboring elements that start from odd and even indices alternatively in each step and apply a differentiable swap operation if elements are in the wrong order. The conditional swap operations on each step s also define a differentiable permutation matrix P_s . The network output is a complete differentiable permutation matrix P , defined as the multiplication of matrices of each step.

defined as:

$$\begin{aligned} d'_i &= \text{softmin}(d_i, d_j) = d_i f(d_j - d_i) + d_j f(d_i - d_j) \\ d'_j &= \text{softmax}(d_i, d_j) = d_i f(d_i - d_j) + d_j f(d_j - d_i) \end{aligned} \quad (3.2)$$

where $f(x) = \frac{1}{\pi} \arctan(\beta x) + 0.5$ [PBKD22]. The hyperparameter $\beta > 0$ denotes an inverse temperature, and when $\beta \rightarrow \infty$ the relaxation converges to the discrete swap operation. The differentiable conditional swap operation for the elements (d_i, d_j) can be defined as a permutation matrix $P_{\text{swap}(d_i, d_j)} \in R^{(K+N) \times (K+N)}$ is an identity matrix except for entries $P_{ii}, P_{ij}, P_{jj}, P_{ji}$, which are defined as:

$$\begin{aligned} P_{ii} &= P_{jj} = f(d_j - d_i), \\ P_{ij} &= P_{ji} = f(d_i - d_j). \end{aligned} \quad (3.3)$$

The permutation matrix P_s for step s is the product of matrices corresponding to parallel (and thus independent) swap operations in this step $P_s = \prod_{i \in R} P_{\text{swap}(d_i, d_{i+1})}$, where R is the set of odd indices if s is odd and the set of even indices if s is even. The complete permutation matrix P is defined as $P = P_{K+N} \cdot \dots \cdot P_1$.

In the discrete case, each row of a permutation matrix has exactly one entry of 1, which indicate the position where the element that corresponds to this row should be placed. In

the relaxed version, row values can be seen as a distribution over possible positions of the element [PKBD22]. For training with sorting supervision, where the correct order of input values is known, we can create a ground truth matrix Q and define the loss as $L = \frac{1}{(K+N)^2} \sum_{i,j} \text{BCE}(P_{ij}, Q_{ij})$ where BCE refers to binary cross-entropy.

3.3.3.2 GroCo Loss

If we would know the ground truth order of positives and negatives, we could create a ground truth permutation matrix and calculate a loss. However, as we rely on random augmentations, we do not know the ground truth order among positives and among negatives. Thus, we propose a solution to obtain a loss to fulfil our group ordering constraints.

Let us again assume that positives are ordered with respect to their distances to the anchor image as $d_1^p \leq \dots \leq d_K^p$ and negatives as $d_1^n \leq \dots \leq d_N^n$. As shown in Figure 3.3, we concatenate positive and negative distances in a list:

$$[d_1^p, \dots, d_K^p, d_1^n, \dots, d_N^n]. \quad (3.4)$$

Even though we have ordered elements in the positive and negative sub-list, we still don't know if the constraint $d_i^p < d_j^n$ is fulfilled for any $1 \leq i \leq K$ and $1 \leq j \leq N$.

Here, we apply a differentiable sorting network to the list and obtain a differentiable permutation matrix for sorting the list in non-descending order. Values in a permutation matrix row can be seen as probabilities to sort the corresponding element to the different positions, e.g., P_{11} would be the probability for assigning the first element in the list (d_1^p) to position 1, P_{12} to position 2, etc. Therefore, a sum of the first K elements in a row can be considered as a probability being sorted inside the first K elements. Thus, for a permutation matrix of size $(K+N) \times (K+N)$ the sum of the first K columns results in probabilities of being sorted in *positive places* and the later columns (from $K+1$ to $K+N$) in *negative places*. We define a loss that enforces positives to be sorted in the positive places and negatives in the negatives places, as

$$L = \frac{1}{2(K+N)} \sum_{i=1}^{K+N} \left(\text{BCE} \left(\sum_{k=1}^K P_{ik}, \mathbb{1}_{i \leq K} \right) + \text{BCE} \left(\sum_{k=K+1}^{K+N} P_{ik}, \mathbb{1}_{i > K} \right) \right) \quad (3.5)$$

where $\mathbb{1}$ is an indicator function.

As illustrated in Figure 3.2, our loss is a relaxation of the sorting supervision that considers only two types of swap operations: swap operations *within* the group of positive and negative samples, which should not contribute to the loss, and swap operations *between* the groups, which violate the positive-negative ordering assumption and which we want to use as the optimization criterion.

Role of Pre-ordering. Practically, we pre-order positives and negatives inside the groups in a non-differentiable way before inputting them into the differentiable sorting network. Though even without it, the loss will still contrast positives to negatives, we believe that pre-ordering fulfills the idea behind GroCO. When we input pre-ordered elements, the sorting network performs comparisons of neighboring elements swapping them if they are in the wrong order, or if everything is sorted correctly, comparing the strongest positive with the strongest negative. In this way, elements are considered as a group, and borders of the groups or their overlapping parts are used in the loss.

Number of Negatives. Since the only strongest negatives (that are sorted incorrectly) mostly participate in the computation of the loss function, we may use only top- N strongest negatives almost without losing any learning signal. We will show in the ablation study that $N = 10$ is enough.

Role of β . The inverse temperature β in differentiable swap operation corresponds to the degree of relaxation of the swap operation that converges to a discrete case when $\beta \rightarrow \infty$ (Figure 3.3). Therefore with lower β the swap operation is more “soft”, which means that the variance in gradients is smaller, which is beneficial for optimization, but a relaxation error accumulated by steps is larger, and vice versa in the case of larger β . With higher β even a small difference between values results in a high probability for a swap or not swap operation, resulting in a smaller margin between the positive and negatives group. With lower β we push positive and negative groups further apart.

3.4 EXPERIMENTAL EVALUATION

3.4.1 Implementation Details

Unless stated otherwise, we use the following setup for all of our experiments:

Augmentation. To create m augmented views per image (considering $m = 2, 3, 4$), we follow the DINO augmentation setup [CTM⁺21]. Since computational cost grows linearly with an increasing number of augmentations, we additionally considered the multi-crop augmentation strategy proposed in SwAV [CMM⁺20]. The idea is to sample low-resolution *local* views along with the standard 224×224 ones. We use $2 \times 224 + 6 \times 96$ scheme, where with two *global* 224×224 augmented views, we also sampled six *local* 96×96 views, giving eight views per image. In this case, we follow “*local-to-global*” correspondence idea [CMM⁺20, CTM⁺21] and use only global views as positives for both local and global anchor images.

Model. To be consistent with previous works [CKNH20, GSA⁺20, CMM⁺20, CH21b] we use Resnet50 [HZRS16] as the encoder $g(\cdot)$ and an MLP block consisting of three fully connected layers with a size of 2048 and followed by a batch normalization layer [IS15] as the projection head $h(\cdot)$. All batch normalization layers except the last one are followed by a ReLU activation. The dimensionality of the representation space and the latent space are both 2048 as in [CH21b]. We also adopt the stop gradient operation, which is widely used in different self-supervised learning methods [CKNH20, CTM⁺21, GSA⁺20]; specifically, we perform stop gradient during distance computation $d(x^a, x_i) = d(x^a, \text{stop_grad}(x_i))$. While we found that it is not strictly necessary for a successful training of our model, we observed that stop gradient makes the training process more stable and allows training the model with a much larger variation of hyperparameters while maintaining stable performance.

Training. Following previous works [CKNH20, GSA⁺20, CMM⁺20, CH21b], we use the train set of the ImageNet ILSVRC-2012 dataset [RDS⁺15] for self-supervised training without any human annotation. We train our model with the SGD optimizer [YGG17] with a learning rate of $24.0 \times (\text{batch size}/1024) \times (100/\text{\#epochs})$ and weight decay of 10^{-6} . By default, we use the top $N = 10$ strongest negative examples and an inverse temperature of $\beta = 1$. Due to resource constraints, we train our model with a batch size of 1024 with mixed precision. We use a 10 epoch linear warm-up and a cosine schedule without restarts [LH16]. Using an 8-GPU (NVIDIA A6000 GPUs) server, training for 100 epochs with $m = 2$ views per image takes approximately 22 hours.

Method	Batch Size	Views	Linear Probing (Top-1)		
			100 ep	200 ep	400 ep
Max-Margin [SSCC22]	256	2×224	63.8	-	-
MoCo v2† [CFGH20]	256	2×224	67.4	69.9	71.0
SimSiam [CH21b]	256	2×224	68.1	70.0	70.8
VICReg [BPL21]	2048	2×224	68.6	-	-
Barlow Twins [ZJM ⁺ 21]	2048	2×224	68.7	-	-
SimCLR† [CKNH20]	4096	2×224	66.5	68.3	69.8
SwAV† [CMM ⁺ 20]	4096	2×224	66.5	69.1	70.7
BYOL† [GSA ⁺ 20]	4096	2×224	66.5	70.6	73.2
Whitening [ESSS21]	4096	4×224	69.4	-	72.6
GroCo (ours)	1024	2×224	69.2	70.4	71.0
GroCo (ours)	1024	4×224	69.6	70.4	71.4
SwAV [CMM ⁺ 20]	4096	$2 \times 224 + 6 \times 96$	72.1	73.9	74.6
GroCo (ours)	1024	$2 \times 224 + 6 \times 96$	71.6	73.0	73.7

Table 3.1: **Comparison with state-of-the-art in linear probing on ImageNet.** We report results for training for 100, 200, and 400 epochs. **Backbone=Resnet50.** †denotes results from SimSiam [CH21b] improved reproductions.

3.4.2 Evaluation Procedure

Linear Probing. First, we follow the standard protocol and evaluated the learned embedding space by linear evaluation [CKNH20, GSA⁺20, CMM⁺20, CH21b], which captures the linear separability of classes in the representation space. For this, we train a linear classifier on frozen representations in a fully-supervised way using the ImageNet train set. We follow the standard protocol [CH21b] to train the linear classifier.

k -NN Evaluation. To analyze local properties of the learned representation, namely how often neighbored data points correspond to the same semantic class, we further evaluate our method by nearest neighbor classification, predicting the object class by a simple weighted k nearest neighbor classifier (k -NN) with $k = \{1, 10, 20\}$ based on cosine distance as in [CTM⁺21, CMM⁺20]. We again use the ImageNet train set for supervision and test on the ImageNet val set.

3.4.3 Comparison to State-of-the-Art

We start with a comparison of the proposed method to state-of-the-art self-supervised learning methods in linear probing and k -NN evaluation on the ImageNet [RDS⁺15] and for image retrieval on the revised Oxford and Paris dataset [RIT⁺18].

Linear Probing. In the case of linear probing (Table 3.1), we observe that our method outperforms almost all baselines in classical (not multi-crop) settings. Our method demonstrates state-of-the-art performance in 100 epochs training, outperforming contrastive baselines SimCLR [CKNH20] and MoCo v2 [CFGH20] by +2% and information maximization methods Barlow Twins and VICReg by +1% in 100 epochs setup. Moreover, our method outperforms SimCLR and clustering-based SwAV [CMM⁺20] in all other settings as well. Also, we even observe competitive performance compared to BYOL [GSA⁺20] and Whitening [ESSS21] methods that use $\times 4$ larger batch size and/or an additional teacher network. We also observe performance improvement by utilizing multiple positive examples during training. In the multi-crop $2 \times 224 + 6 \times 224$ settings, our method is on par with clustering-based SwAV, which

Method	Batch Size	Views	k -NN (weighted, $k=20$)		
			100 ep	200 ep	400 ep
MoCo v2 [CFGH20]	256	2×224	-	55.6	-
SimSiam [CH21b]	256	2×224	57.4	-	-
SimCLR [CKNH20]	4096	2×224	53.8	57.2	59.2
SwAV [CMM ⁺ 20]	4096	2×224	-	-	61.3
GroCo (ours)	1024	2×224	60.5	62.7	63.6
GroCo (ours)	1024	4×224	61.8	63.6	65.0
SwAV [CMM ⁺ 20]	4096	$2 \times 224 + 6 \times 96$	61.7	63.7	64.9
GroCo (ours)	1024	$2 \times 224 + 6 \times 96$	62.2	64.4	65.2

Table 3.2: **Comparison with state-of-the-art in k -NN performance on ImageNet.** We evaluate k -NN performance (Top-1) on officially released weights of baseline models trained for 100, 200, and 400 epochs. We used a simple weighted k -NN with $k=20$. **Backbone=Resnet50.**

Method	Epochs	Batch Size	Views	Oxford		Paris	
				M	H	M	H
SimSiam [CH21b]	100	256	2×224	26.89	7.04	46.92	19.31
MoCo v2 [CFGH20]	200	256	2×224	23.28	5.07	42.8	17.33
SimCLR [CKNH20]	400	4096	2×224	23.27	4.56	46.93	20.19
SwAV [CMM ⁺ 20]	400	4096	2×224	28.01	8.35	46.23	17.4
GroCo (ours)	400	1024	2×224	31.77	9.08	54.79	25.77

Table 3.3: **Comparison with state-of-the-art in image retrieval.** We evaluate image retrieval performance on Medium (M) and Hard (H) splits of revisited Oxford and Paris datasets [RIT⁺18]. We evaluate nearest neighbor retrieval performance with ImageNet-trained encoders and report Mean Average Precision.

utilizes $\times 4$ larger batch size.

k -NN Evaluation. We further evaluate the k -NN performance of our methods compared to multiple state-of-the-art methods with officially released weights in Table 3.2. We observe that the proposed method excels in all settings, outperforming all methods by more than 4% in the no multi-crop scenario. The method further greatly improves performance by leveraging multiple positive examples. Finally, by utilizing local-to-global correspondences our method outperforms the SwAV method that uses $\times 4$ larger batch size.

Image Retrieval. To further demonstrate the potential of the proposed methods in nearest neighbors-based tasks, we additionally evaluate ImageNet-trained self-supervised learning methods in image retrieval on the revisited Oxford and Paris datasets [RIT⁺18] in Table 3.3. We report the Mean Average Precision for the Medium (M) and Hard (H) splits of the datasets as in [CTM⁺21]. We observe that our method outperforms all other methods in this task, confirming its good local properties of learned representations.

3.4.4 Comparison to the Pairwise Contrastive Loss

To evaluate the properties of the proposed method in a direct comparison with the pairwise contrastive loss formulation, we compared our method to the classical contrastive learning method SimCLR [CKNH20], which uses the most popular InfoNCE loss [OLV18] in training. Since SimCLR originally uses only two augmentations per image, we extend it to a group of positives scenario by applying contrastive loss between all possible positive pairs (see Ap-

Method	Views	k -NN Evaluation			Linear Eval.	
		$k=1$	$k=10$	$k=20$	Top-1	Top-5
SimCLR	2×224	-	-	-	64.3	-
SimCLR†	2×224	46.0	51.5	51.9	65.7	86.7
SimCLR†	3×224	44.7	50.0	50.6	65.8	86.8
SimCLR†	4×224	46.3	52.1	52.6	66.5	87.1
SimCLR†	$2 \times 224 + 6 \times 96$	46.6	51.4	52.0	67.2	87.7
GroCo (ours)	2×224	55.3	60.3	60.5	69.2	88.4
GroCo (ours)	3×224	55.8	61.2	61.6	69.5	88.8
GroCo (ours)	4×224	56.4	61.5	61.8	69.6	88.9
GroCo (ours)	$2 \times 224 + 6 \times 96$	57.0	61.9	62.2	71.6	90.4

Table 3.4: **Comparison with SimCLR as a contrastive baseline on ImageNet.** Results are reported for k -NN and linear probing. The best results are bolded. **Backbone=Resnet50, #epochs=100, batch size=1024.** †denotes our reproduction.

Method	Evaluation	Aircraft	Caltech101	Cars	Cifar10	Cifar100	DTD	Flowers	Food	Pets	SUN397	VOC2007	Average
SimCLR†	k -NN (weighted, $k=20$)	19.5	75.8	15.2	85.1	63.3	70.9	73.1	47.9	60.6	46.3	70.0	57.06
GroCo (ours)	k -NN (weighted, $k=20$)	23.7	79.3	17.5	85.4	63.6	69.3	77.7	52.8	75.9	50.1	73.7	60.81
SimCLR†	Linear transfer	42.51	85.10	43.19	90.86	74.54	75.11	91.59	68.20	77.92	58.43	78.74	71.47
GroCo (ours)	Linear transfer	45.90	86.44	42.63	89.26	70.72	72.77	88.41	66.72	84.29	58.51	81.34	71.55

Table 3.5: **Comparison with SimCLR as a contrastive baseline in transfer performance on 11 classification datasets.** Models are pretrained on Imagenet. **Backbone=Resnet50, #epochs=100, batch size=1024, views=2x224.** †denotes our reproduction.

pendix A.5). For a fair comparison, we reproduced SimCLR with the 3-layers MLP projection head (as in our method) [CKS⁺20].

Representation Learning Performance. First, we evaluated the learned representations based on the numbers of positive samples in Table 3.4. We consider four scenarios. First, in the 2×224 scenario, we sample only two augmentations per image, and therefore we have only one positive example per data point. In this case, our loss works similarly to SimCLR in a way that it is minimizing the distance between the positive example and the anchor, however, our loss still considers negatives as a group and utilizes only the nearest negative elements that are sorted incorrectly, therefore focusing on local properties of representation space. While outperforming SimCLR baseline by +3% (Top-1) in linear probing, our method also advances in k -NN evaluation by more than +10% ($k=20$), demonstrating that our loss helps to learn better representation not only in terms of linear separability but also in terms of local structure. We further consider settings with 3×224 , 4×224 , and $2 \times 224 + 6 \times 96$ views. We observe that SimCLR shows mixed results from utilizing more views: while it benefits in linear evaluation, performance is not stable in k -NN, for 3×224 performance is lower and for $2 \times 224 + 6 \times 96$ is the same. However, our method clearly profits from utilizing more positives in both evaluations, resulting in +1.7% in k -NN +2.4% in linear evaluation for $2 \times 224 + 6 \times 96$ setup.

Transfer Performance. We further evaluated how well performance transfers on other datasets. In Table 3.5 we compare Imagenet pretrained models in k -NN evaluation and linear transfer and on 11 many-shot classification datasets, including FGVC Aircraft [MRK⁺13], Caltech-101 [FFFPo4], Stanford Cars [KDSFF13], CIFAR10 [KH⁺09], CIFAR-100 [KH⁺09], DTD [CMK⁺14], Oxford 102 Flowers [NZo8], Food-101 [BGG14], Oxford-IIIT Pets [PVZ]12],

	k -NN Evaluation			Linear Probing	
	$k=1$	$k=10$	$k=20$	Top-1	Top-5
Triplet Loss (margin=0.8)	46.8	52.5	52.8	63.9	85.4
Triplet Loss (margin=1.6)	47.9	53.4	53.7	64.2	85.3
Triplet Loss (margin= $+\infty$)	47.9	53.3	53.8	64.3	85.3
GroCo (Ours)	54.7	59.5	60.0	68.5	88.2

(a) Ordering supervision.

	k -NN Evaluation			Linear Probing	
	$k=1$	$k=10$	$k=20$	Top-1	Top-5
Randomly ordered	54.1	59.2	59.4	68.4	88.2
Pre-ordered (Ours)	54.7	59.5	60.0	68.5	88.2

(b) Pre-ordering in the groups.

Inverse temp. β	Number of negatives N					
	5		10		20	
	k -NN	Lin.p.	k -NN	Lin.p.	k -NN	Lin.p.
0.25	58.9	67.9	-	-	-	-
0.5	59.1	67.5	59.8	68.5	-	-
1	58.8	67.7	60.0	68.5	54.4	65.2
2	-	-	60.0	68.3	56.2	65.7
4	-	-	-	-	59.2	67.7
8	-	-	-	-	58.6	67.7

(c) An inverse temperature β , a number of negatives N .

Batch Size	k -NN Evaluation			Linear Probing	
	$k=1$	$k=10$	$k=20$	Top-1	Top-5
256	52.2	56.8	56.8	67.2	87.7
512	53.1	57.9	58.1	67.7	87.8
1024	54.7	59.5	60.0	68.5	88.2

(d) Batch size.

Table 3.6: **Ablation Experiments.** For (c), we report k -NN performance with $k=20$, and linear probing performance (Top-1), denoted as Lin.p. The best results are bolded. Options used to obtain the main results are highlighted. **Backbone=Resnet50, Views= 2×224 , #epochs=100, Weight Decay=0.5e-6.**

SUN397 [XHE⁺10] and Pascal VOC2007 [EVGW⁺10]. For linear transfer evaluation, we follow the experimental protocol provided in [EGH21] without any modification. We observe that the proposed method is on par with SimCLR in linear evaluation. In zero-shot k -NN evaluation, our method excels in 10 out of 11 datasets, in particular improving performance on the Pets dataset by +15%, and on the Aircraft and Food datasets by +4%. This shows that the improved k -NN performance transfers to downstream tasks as well.

3.4.5 Ablation Study

Finally, we analyze the design choices of our method.

Ordering Supervision. First, we compare the group ordering supervision via differentiable sorting with a triplet loss formulation $L = \max(d_i^p - d_j^n + r, 0)$, where r is a margin. For a

fair comparison, we consider all positive and the 10 strongest negative samples and evaluate different margin parameters in Table 3.6a. Here, sorting is superior to a triplet loss with hard margin selection.

Pre-ordering in Groups. We analyze the impact of pre-ordering elements within the negative and positive groups before forwarding them to the sorting network in Table 3.6b. We observe that our methods achieve competitive performance even without pre-ordering; however, pre-ordering strengthens our method further.

Inverse Temperature, Number of Negatives. In Table 3.6c, we examine the influence of the number of nearest neighbor negatives N used in the loss, as well as the value of the inverse temperature parameter β (Equation 3.2). We observe that usage of too many negatives may not be beneficial for the model. Since our loss focuses on negatives that are sorted incorrectly, increasing the number of negatives at some point does not bring any new learning signal (because more distant is unlikely to be sorted incorrectly). However, a larger N results in more steps of the sorting network, increasing the degree of relaxation. Using a larger inverse temperature β (leading to a lower degree of relaxation in the swap operation) we can gain some performance; however, the variance of the gradients is larger with a larger β , which is not beneficial for optimization. We found that $N = 10$ and $\beta = 1$ is the most efficient configuration.

Batch Size. The large batch size might be an important factor in obtaining good performance for many self-supervised learning methods. We find that our method is quite robust for training with smaller batch sizes (Table 3.6d).

Training Time. We also consider the training time of our model compared to SimCLR baseline. To eliminate the influence of distributed training, we measure the average time of training iteration on one GPU. We find that the iteration time of both models is comparable, 514ms for SimCLR vs 526ms for the proposed methods for a batch size of 128.

3.5 CONCLUSION

In this chapter, we proposed an alternative approach to the common pairwise contrastive learning formulation commonly used in self-supervised learning and multimodal learning. Namely, we proposed the group ordering constraints, that consider positives and negatives as groups and enforce the group of positives to be closer to the anchor image than the negative group. To enforce these constraints, we leveraged recent progress in the context of differentiable sorting approaches and formulated a group ordering loss based on the given sorting supervision. We evaluated our new loss function for image representation learning from unlabeled images, and our results showed that the proposed framework not only competes with current contrastive loss baselines but also outperforms standard contrastive learning across all k -NN-based metrics. In Chapter 4, we continue to explore pretraining with unsupervised data, this time focusing not on unimodal but on multimodal representations that can combine multiple input modalities into a single *fused* representation to enhance visual recognition with complementary information.

EVERYTHING AT ONCE – MULTI-MODAL FUSION TRANSFORMER FOR VIDEO RETRIEVAL

Contents

4.1	Introduction	44
4.2	Related Work	46
4.2.1	Multi-modal learning	46
4.2.2	Multi-modal learning with transformers	46
4.3	Method	47
4.3.1	Problem Statement	47
4.3.2	Model Architecture	47
4.3.3	Combinatorial Loss	49
4.4	Experimental Evaluation	50
4.4.1	Experimental Setup	50
4.4.2	Datasets, Tasks, and Metrics	51
4.4.3	Comparison to State-of-the-Art	52
4.4.4	Ablation Study	53
4.5	Conclusion	56

This chapter explores learning strong representations for video understanding that are able to fuse multiple complementary modalities, such as video, audio, and text, into a joint embedding, training solely on freely available instructional videos without relying on human-annotated data. Multi-modal learning from video data has recently received increased attention, as it enables training semantically meaningful embeddings without manual annotation, supporting tasks such as zero-shot retrieval and action localization. In this work, we present a multi-modal, modality-agnostic fusion transformer that learns to exchange information between multiple modalities and integrate them into a fused representation within a shared multi-modal embedding space, thereby yielding representations that can exploit complementary information for video understanding. We propose training the system with a combinatorial loss on *everything at once* – any combination of input modalities, including single modalities as well as modality pairs, while explicitly leaving out add-ons such as position or modality encodings. At test time, the resulting model can process and fuse any number of input modalities. Moreover, the implicit properties of the transformer allow it to handle inputs of different lengths. To evaluate the proposed approach, we train the model on the large-scale HowTo100M dataset of instructional videos collected from the internet and assess the resulting embedding space on four challenging benchmark datasets, achieving state-of-the-art results in zero-shot video retrieval and zero-shot action localization.

This chapter is based on the CVPR 2022 paper [SCR⁺22]. Nina Shvetsova, as the first author, contributed to the design of the proposed method, implemented it, conducted all experiments and analyses, and was the main writer of [SCR⁺22]. This work has been cited more than 180 times and has been used as a foundation for developing further methods [CSR⁺24, SRS⁺23, SKH⁺24, SKSK23].

4.1 INTRODUCTION

Humans capture their world in various ways, combining different sensory input modalities such as vision, sound, touch, and more, to make sense of their environment. Video data approximates this type of input by combining visual and audio information as two coherent and complementary signals that can be further enhanced with a text description. Recent research has therefore started to explore how the information of those different modalities can be leveraged to learn meaningful representations from this kind of content. Such systems can be used for representation learning, for example, to learn multi-modal embedding spaces on video data [ARS⁺20, AYQ⁺21], where the input of one modality such as text, can be matched to one or more other modalities such as video and audio, enabling tasks such as nearest-neighbor based zero-shot classification or video retrieval [MZA⁺19, GSAS20, RBH⁺21]. Our work in this chapter focuses on the later problem, namely the learning of meaningful multi-modal embedding spaces. Current approaches in this area usually learn encodings for different modalities by projecting inputs to a common space and applying contrastive loss to bring embeddings from co-occurred modalities together. Such approaches can be based on classical neural network elements to learn those encodings [MZA⁺19, RBH⁺21, CRD⁺21, MAS⁺20, ABARB20], i.e. convolutional neural networks backbones and non-linear projections [MZA⁺19], multiple instance learning [MAS⁺20], or clustering [CRD⁺21]. More recently transformer based methods have also been proposed [GSAS20, BNVZ21a, AYQ⁺21, LJZ⁺21]. To generate the final embedding space, they use multiple independent single-modality self-attention transformer blocks [BNVZ21a, LJZ⁺21, GRHD22], or a single transformer model for all modalities [GSAS20], or a single modality-agnostic transformer [AYQ⁺21]. In the last approach, modalities are still processed independently and one-by-one forwarded to achieve a single-modality embedding. But so far, none of these transformers allow for adaption to any given number of input modalities. Although modality-agnostic transformers that handle multiple input modalities such as PerceiverIO [JBA⁺21] have been proposed, they have been constructed for learning a latent space that can cover multiple tasks in different domains. Compared to our work, the latent space in such cases mainly serves the purpose of compressing multiple inputs and tasks in one model.

In this work, we propose an approach that leverages self-attention for multi-modal learning which jointly processes any number of modalities and allows modalities to attend to each other. A high level overview our architecture is shown in Figure 4.1. Input tokens from one or more modalities are passed through a fusion transformer that attends features relevant for a combined input, followed by a projection to a joint multi-modal embedding space. We design and train the fusion transformer to cover three aspects of multi-modal video learning: first, it should allow modalities to attend to each other and learn multi-modal correlations; second, it should be modality-agnostic and handle any possible modality input combination; and third, as different modalities and samples can vary in length, it should be able to process input of any length. To enable the fusion transformer to address all those tasks, we follow the idea of a universal self-attention in the transformer block and share key, query, and value weights to all tokens, agnostic of their input modality. In this way, self-attentions learns which input tokens to attend from single modalities as well as from any combination of modalities in a general way.

To train the model, we propose a combinatorial loss function which considers contrastive loss between all possible and available input combinations. For example, in the case of vision, text, and audio, the loss is based on each modality embedding alone as well as based on pairwise vision-text, audio-text, and text-audio combinations as shown in Figure 4.1. The

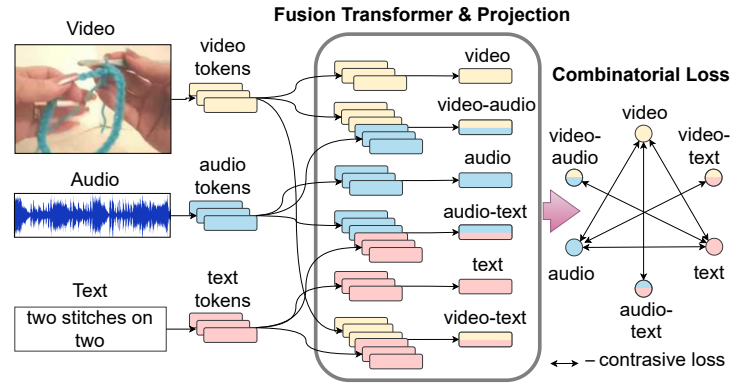


Figure 4.1: Overview of the proposed approach for self-supervised learning of multi-modal embedding space. The fusion transformer is able to process any combination of input modalities. Internally, the transformer allows each modality to attend to each other. The proposed architecture is trained with a combinatorial contrastive loss considering each possible combination of input modalities.

resulting model is thus able to fuse any number of input modalities at test time. Compared to other universal self-attention methods, we omit any meta information encoding such as position or modality embedding. This further allows us to process any input of different lengths, as we are no longer bound to a maximum input size defined at training time. Note that while we refer to this transformer as a fusion transformer, we are not proposing a new transformer architecture, but rather refer to it as a transformer that is trained in a way that enables fusion without any need for changes to the self-attention mechanism. As a result, the final modal can be used for any type of input, single modalities or combinations of multiple ones, as well as for any input length.

We evaluate the proposed approach by training the model on the HowTo100M dataset [MZA⁺19] and testing its zero-shot text-to-video retrieval and step action localization on four downstream datasets, namely YouCook2 [ZXC18], MSR-VTT [XMYR16], CrossTask [ZAC⁺19b] and Mining YouTube [KIRG19]. Our results show that the proposed combination of a fusion transformer together with a combinatorial loss function improves performance and leads to new state-of-the-art results. We summarize the contributions of this work as follows:

- We propose a multi-modal fusion transformer that processes input of any combination of modalities and any length and attends relevant features with respect to cross-modal information.
- We propose a combinatorial contrastive loss that considers all possible combinations of input modalities at training time.
- We show that using such a multi-modal fusion transformer as an intermediate processing step can significantly improve performance for multi-modal embedding space learning.

4.2 RELATED WORK

4.2.1 Multi-modal learning

The idea of learning from more than one modality can be seen as an integral part of machine learning research, comprising areas such as vision-language learning [RKH⁺21c, ZLH⁺21b], vision-audio learning [CXVZ20a, XLG⁺21, TKM20, AZ18, AZ17, AVT16, HRS⁺18], zero-shot learning [HE20, MNXA21], cross-modal generation [ZWF⁺18, MMS19, RAY⁺16], as well as multi-modal multi-task learning [KGS⁺17]. Video naturally combines multiple modalities, while at the same time allowing to learn from large-scale data that would not be annotatable in a reasonable time. In this context, Miech et al. [MZA⁺19] proposed the HowTo100M dataset of narrated videos and presented a system showing the potential of multi-modal learning for learning a video-text embedding space via contrastive loss. The dataset contains YouTube instructional videos that come with audio and respective subtitles as a textual description obtained by Automatic Speech Recognition (ASR). As this data can be considered more noisy than curated vision-text datasets, Amrani et al. [ABARB20] proposed a noise estimation for multi-modal data via multi-modal density estimation. Miech et al. [MAS⁺20] proposed MIL-NCE, combining the idea of noise-contrastive estimation with a multiple instance learning formulation. Alwassel et al. [AMK⁺20] used the audio and video information only and proposed to leverage unsupervised clustering as a supervisory signal across modalities. While those works [MZA⁺19, ABARB20, MAS⁺20, AMK⁺20] only use two modalities to train their models, others have focused on the problem of learning from vision, audio, and text at once [AVT17, ARS⁺20, DCS⁺21, RBH⁺21, CRD⁺21]. As perhaps one of the first, Aytar et al. [AVT17] proposed an architecture trained on image-text and image-audio pairs that allows to connect text and audio modalities. Later Alayrac et al. [ARS⁺20] followed the idea of different embedding spaces for different modality combinations and proposed Multi-Modal Versatile Networks. A shared embedding space was proposed by Rouditchenko et al. [RBH⁺21] mapping all three modalities in one joint space. This idea has recently been extended by additional clustering and reconstruction loss by Chen et al. [CRD⁺21].

4.2.2 Multi-modal learning with transformers

Architectures based on self-attention and transformers have been explored to learn from multi-modal video data. Cheng et al. [CWP⁺20] proposed a co-attention module to learn correspondences between audio and video samples. Luo et al. [LJS⁺20a] pick up on that idea but proposed, similar to Uniter [CLY⁺20] for vision-language tasks, a joint cross-modal encoding for video-text pairs. Compared to that, Bain et al. [BNVZ21a] focused on the problem of how to attend to temporal as well as spatial information in the video backbone. They therefore processed both modalities, video and text, in two separate transformer backbones and only added a linear mapping layer on top of the backbones. In this context, recently, Nagrani et al. [NYA⁺21] proposed a multi-modal bottleneck transformer for effective audio-visual fusion trained in the supervised setting. A transformer-based approach that actually uses all three modalities, and can therefore be considered closest to our proposed work, has been proposed by Akbari et al. [AYQ⁺21]. Here, a single backbone transformer is applied to any of the modalities separately, but with shared attention. For training, the model follows the idea of [ARS⁺20] and computes the matching of video-audio first, followed by video-text matching. It thus fuses those modalities in a pairwise way, which can be compared to a

subset of our proposed loss function. Other approaches also leverage temporal aspects in context of multi-modal transformer learning. Gabeur et al. [GSAS20] used a combination of expert and temporal embeddings to train a multi-modal transformer while Wang et al. [WZY21a] proposed a local-global temporal alignment based on multi-modal experts to guide the training. The idea of simply using a pretrained vision-language transformer model has also been explored by Lou et al. [LJZ⁺21], using the pretrained CLIP model [RKH⁺21c] as a backbone with a transformer-based similarity encoder on top of a vision and text backbone and achieving good results on tasks such as video retrieval. As most transformer-based method use various and sometimes not-publicly available datasets for backbone pretraining or have a need for resources that make it hard to repeat experiments, it is difficult to directly compare performance across different architectures and pretraining set. We therefore decided to follow the setup used in majority of works here and rely on pre-extracted features that are then processed by the proposed architecture to allow a direct comparison with previous works.

4.3 METHOD

Our goal is to learn a projection function of single modalities or a set of modalities into a joint embedding space in a way that semantically similar inputs would be close to each other, e.g. the projection of the text description of a video scene should be close to the projection of the video-audio representation of this scene. In the following, we consider three modalities, video, audio, and text (corresponding ASR caption or linguistic narration); but the proposed method can be extended to more modalities.

4.3.1 Problem Statement

Given a set of text-video-audio triplets $\{(t_i, v_i, a_i)\}_{i=1}^N \in (\mathcal{T} \times \mathcal{V} \times \mathcal{A})^N$ of N video clips obtained from the data distribution, we are learning a projection $f(\cdot, \cdot, \cdot)$ that can take up to three inputs: text t , video v , and audio a and produce d -dimensional embedding representation of the input. For the simplicity of the notation, we will omit missing modalities, so that $f(t, v)$ will stand for projection $\mathcal{T} \times \mathcal{V} \rightarrow \mathbb{R}^d$ and represent the joint embedding of text t and video v . Our goal is to maximize the dot-product similarities between semantically related inputs $f(t), f(v), f(a), f(t, a), f(t, v), f(v, a)$ (such as when t, v , and a are from the same video clip) and minimize otherwise.

4.3.2 Model Architecture

4.3.2.1 Token Creation

As illustrated in Figure 4.2, our architecture starts from features extracted from modality-specific backbones. We transform sets of extracted feature vectors into token space by learnable modality-specific projections and modality-specific normalization layers [BKH16]. As a result, for the (t_i, v_i, a_i) input triplet, we obtain three sets of tokens: $[\tau_{i_1}, \dots, \tau_{i_k}]$ from text t_i , $[\nu_{i_1}, \dots, \nu_{i_m}]$ from video v_i , $[\alpha_{i_1}, \dots, \alpha_{i_n}]$ from audio a_i . As the number of tokens may vary, e.g. depending on the length of the video clip, we normalize the length of inputs per batch to allow batch processing by padding and using attention masks [VSP⁺17]. Practically, for comparability, we follow the protocol of [MZA⁺19, RBH⁺21] and train the model on fixed-length video clips. Technically, the model can handle clips of variable length, also with respect to different

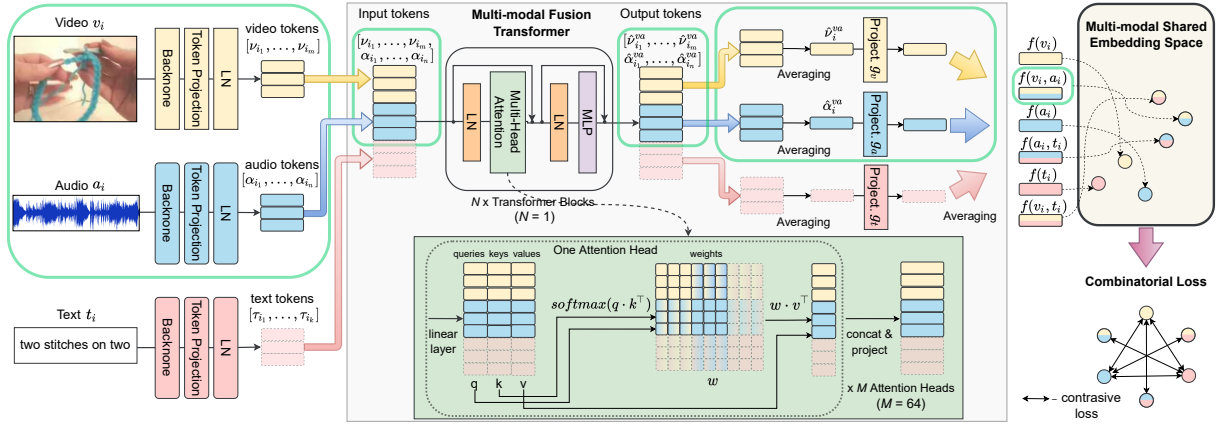


Figure 4.2: Schematic visualization the proposed method. While tokens from different modalities are processed in all possible combinations, we exemplarily consider the video-audio pair marked with green rectangles here. The input tokens are forwarded together through the fusion transformer layer and attended by the respective weights, which are based on the combinations of keys and queries of input tokens in various modalities. The resulting outputs of multiple heads are then concatenated and projected to the final token space, which is then used to project each modality separately into the joint embedding space. During training, we apply the model six times to obtain six embeddings corresponding to text, video, audio, text-video, text-audio, and video-audio modalities to compute the combinatorial loss.

modalities, at training and at test time if needed.

4.3.2.2 No Positional Embeddings

Unlike other transformer-based methods [CLY⁺20, BNVZ21a, AYQ⁺21, LLZ⁺21a, TLB21], we omit adding any positional or type embedding information to the tokens. The reason for this is three-fold. With respect to type embedding, it can be assumed that tokens already encode this information based on the fact that they are generated by different backbones, and hence each come with their own “fingerprints.” Positional information has been shown to be beneficial in the context of consistently structured data such as sentences. But in the case of multi-modal video learning, clips are sampled randomly from a larger video sequence at training time, usually without considering shot boundaries or speech pauses. We therefore do not expect a consistent temporal pattern in the sense that a clip always starts at the beginning of an action. Thus leaving out positional embedding might prevent adding noise during training. At inference time, avoiding the positional embedding allows us to process sequences longer than used in the training.

4.3.2.3 Multi-modal Fusion Transformer

As our goal is to learn the representations of any number and combination of input modalities, we want the projection f to learn how to fuse information from multiple modalities to enhance the joint embedding representation. For this purpose, we propose a multi-modal, modality agnostic transformer, where the keys, queries, and values of the input tokens and all further transformations are computed independently from the modality. To create our multi-modal fusion transformer, we adopt regular transformer blocks [VSP⁺17]. Each transformer block

consist of a multiheaded self-attention and a multilayer perceptron (MLP) with two LayerNorm (LN) transforms before them along with two residual connections, as illustrated in Figure 4.2. Note that the difference compared to other methods is not in the architecture itself, but in the way it is trained and the fact that the resulting fusion can actually be learned by a vanilla transformer block, if it is specifically trained for this task. Fusion transformer thus refers to the way a transformer block can be used rather than to a new architecture.

We train the system with a combinatorial input. Namely, we apply it to joint sets of input tokens from all possible combinations of modalities: singles - t , v , a and pairs - (t, v) , (v, a) , (t, a) , allowing tokens from one modality to attend to tokens of other modalities. In this way, we can obtain a *fused* representation from multiple modalities: the combination (t, v) will result in a fused representation of text and video modalities denoted as tv , resp. for va - video and audio, and ta - text and audio. Note that e.g. in case of four modalities, we would consider all combinations up to a triplet (t, v, a) during training. As more modalities would be added, the number of combinations would grow to the point where it might be infeasible to consider all configurations. In this case random modality dropout could be used during training as done in AVSlowfast [XLG⁺21] or Perceiver [JBA⁺21].

Since we want the fusion transformer to be modality agnostic, in each training iteration, we apply it six times to obtain six representations for each sample i : t_i , v_i , a_i , $t_i v_i$, $v_i a_i$, $t_i a_i$. To obtain each representation, we create a joint list of tokens, e.g. for $v_i a_i$: $[v_{i_1}, \dots, v_{i_m}, a_{i_1}, \dots, a_{i_n}]$. We apply the transformer to this input and obtain output tokens as e.g. $[\hat{v}_{i_1}^{va}, \dots, \hat{v}_{i_m}^{va}, \hat{a}_{i_1}^{va}, \dots, \hat{a}_{i_n}^{va}]$ for $v_i a_i$ (with superscript va denoting that tokens were attended to both v and a modalities), where each token was attended with information from another tokens. Note that, unlike the ViT model [DBK⁺20], we do not prepend a learnable $[cls]$ token, which usually serves as a joint representation of all tokens. In our ablation studies we show that this is beneficial for the model (Section 4.4.4).

4.3.2.4 Projection to Shared Embedding Space

With the resulting output tokens, we create the final embeddings for each modality. For each training sample, we get six output sets of tokens and thus embeddings. As an example, we consider the case of creating the representation for $v_i a_i$. We divide output tokens $[\hat{v}_{i_1}^{va}, \dots, \hat{v}_{i_m}^{va}, \hat{a}_{i_1}^{va}, \dots, \hat{a}_{i_n}^{va}]$ into groups based on modality: $[\hat{v}_{i_1}^{va}, \dots, \hat{v}_{i_m}^{va}]$ and $[\hat{a}_{i_1}^{va}, \dots, \hat{a}_{i_n}^{va}]$ and then average them: $\hat{v}_i^{va} = \sum_{j=1}^m \hat{v}_{i_j}^{va}$, $\hat{a}_i^{va} = \sum_{j=1}^n \hat{a}_{i_j}^{va}$. As a result, we obtain a vector representation for each modality included in this computation. But since modalities, even when enhanced with other modalities, are still very different, we project them into the shared embedding space by the learnable modality-specific projections g_t , g_v , or g_a for projections for t , v , a respectively, normalize them, and then combine into a *final embedding vector*:

$$f(v_i, a_i) = \text{norm}(\text{norm}(g_v(\hat{v}_i^{va})) + \text{norm}(g_a(\hat{a}_i^{va}))). \quad (4.1)$$

The normalization ("norm") is used to align the magnitude of vectors. When computing dot product similarity, we take into account only the angle between vectors.

4.3.3 Combinatorial Loss

Contrastive loss can be used to learn representations such that semantically similar inputs are mapped close to each other. Unlike other methods [CRD⁺21, AYQ⁺21, ARS⁺20, RBH⁺21] that learn how to bring modalities together by training with three pairwise single-modality contrastive losses, L_{t-v} between (t, v) , L_{t-a} between (t, a) , and L_{v-a} between (v, a) , we force

tokens to exchange information between modalities while enabling additional contrastive losses: L_{t_va} between (t, va) , L_{v_ta} between (v, ta) , and L_{a_tv} between (a, tv) , and introduce our *combinatorial loss*:

$$L = \lambda_{t_v} L_{t_v} + \lambda_{v_a} L_{v_a} + \lambda_{t_a} L_{t_a} + \lambda_{t_va} L_{t_va} + \lambda_{v_ta} L_{v_ta} + \lambda_{a_tv} L_{a_tv}, \quad (4.2)$$

where $\lambda_{m_{\hat{m}}}$ denotes a weighting coefficient of $(m, \hat{m})$.

Our combinatorial loss considers all possible and available modality combinations and can be generalized to any set of modalities $\mathcal{M} = \{m_1, \dots, m_N\}$ as follows:

$$L = \sum_{\mathcal{X}, \mathcal{Y} \subset \mathcal{M}; \mathcal{X} \cap \mathcal{Y} = \emptyset} \lambda_{\mathcal{X}_Y} L_{\mathcal{X}_Y}, \quad (4.3)$$

where $L_{\mathcal{X}_Y}$ is a contrastive loss between the fused representations of subsets $\mathcal{X}$ and $\mathcal{Y}$, $\lambda_{\mathcal{X}_Y}$ is a weighting coefficient.

To compute the contrastive losses for all combinations, we utilize Noise Contrastive Estimation [OLV18] with temperature τ and batch size B :

$$L_{\mathcal{X}_Y} = -\frac{1}{B} \sum_{i=1}^B \log \left(\frac{\exp(x_i^\top y_i / \tau)}{\sum_{j=1}^B \exp(x_i^\top y_j / \tau)} \right) - \frac{1}{B} \sum_{i=1}^B \log \left(\frac{\exp(x_i^\top y_i / \tau)}{\sum_{j=1}^B \exp(x_j^\top y_i / \tau)} \right). \quad (4.4)$$

By combining both aspects, the processing of all possible modality combinations and the training of the system with the proposed combinatorial loss, we obtain a multi-modal fusion transformer that learns how to attend tokens from one modality to the tokens from all other modalities.

4.4 EXPERIMENTAL EVALUATION

4.4.1 Experimental Setup

If not stated otherwise, we use the following experimental setup in all our experiments and ablation studies.

Backbones. To ensure comparability, we follow the setup of previous works [MZA⁺19, RBH⁺21, CRD⁺21, ABARB20] which is as follows: as visual backbone, we use a combination of ResNet-152 [HZRS16], pretrained on Imagenet [DDS⁺09] and compute one 2D-feature (2048-dimensional vector) per second, as well as ResNeXt-101 [HKS18] pretrained on Kinetics [CZ17] to get 1.5 3D-feature (2048 dim.) per second. We temporally upsample 2D-features with nearest neighbors to have the same number of features as 3D-features and concatenate them to obtain 4096-dimensional vectors. As a text backbone, GoogleNews pretrained Word2vec model [MCCD13] is used with 300-dimensional embedding per word. These backbones are fixed and not finetuned during training. Following [RBH⁺21, CRD⁺21], we use a trainable CNN with residual layers as an audio backbone and adapt the last two residual blocks to extract 1.5 4096-dimensional features per second (see Appendix B.3).

Data Sampling. We use a batch of 224 videos and randomly sample ten 8-second clips per video. If the sampled clip contains narration (95% of all clips), we use ASR time stamps to select clip borders. To disentangle the very high text-audio correlation in HowTo100M, and to avoid text being learned just as an audio narration, we shift the audio clip randomly by 4 seconds with respect to the video and text boundaries.

Projections. Following [RBH⁺21, CRD⁺21, MZA⁺19], we use a gated linear projection [MLS17] to project features into common token space, as well as to project resulting tokens into shared embedding space. We set the dimension of the common token space to 4096 and of the shared embedding space to 6144.

Transformer architecture. As a multi-modal fusion transformer, we use one transformer block with a hidden size of 4096, 64 heads, and an MLP size of 4096.

Loss computation. We use a temperature of 0.05 in NCE and normalize vectors prior to computing dot products. Since not every video clip has all three modalities, we computed NCE only over non-empty embeddings. Following [ARS⁺20], we set a larger weight for a text-visual loss in Equation 4.2, since it was beneficial for training on HowTo100M: $\lambda_{t_v} = 1$, $\lambda_{v_a} = \lambda_{t_a} = \lambda_{t_{va}} = \lambda_{v_{ta}} = \lambda_{a_{tv}} = 0.1$.

Optimization. We train all models for 15 epochs using an Adam optimizer [KB15a] with a learning rate of $5e-5$ and exponential decay of 0.9.

4.4.2 Datasets, Tasks, and Metrics

Pretraining Dataset. We train our model on the HowTo100M dataset [MZA⁺19], which contains over 1 million instructional videos with automatically generated text narrations. The text narrations can be assumed to be noisy and to not always describe the video scene [MZA⁺19].

Zero-shot Text-to-video Retrieval. We use MSR-VTT [XMYR16] and YouCook2 [ZXC18] datasets to evaluate the zero-shot text-to-video retrieval capability of our model. The YouCook2 dataset contains instructional cooking videos from YouTube with human-annotated clips ($\sim 2 - 200$ seconds). For evaluation we use at maximum first 48 seconds of video clip, since most video clips are shorter than that. The MSR-VTT dataset contains human-annotated video clips ($\sim 10 - 30$ seconds) on various topics and provides captions with natural language sentences. Following [MZA⁺19, RBH⁺21, CRD⁺21, MAS⁺20], to evaluate our model on MSR-VTT, we use the 1k set of test clips [YKK18], and for YouCook2, we use 3,350 validation clips [MZA⁺19]. To perform retrieval, we compute similarities by dot product between a text query t and all videos in the dataset using a fused va representation for each video. We report standard recall metrics $R@1$, $R@5$, $R@10$ and the median rank (MedR).

Text-to-video Retrieval after finetuning. We additionally evaluate the retrieval performance of the models finetuned on downstream tasks. Following [RBH⁺21], we used 9,586 training clips to finetune the model on the YouCook2 dataset, and 6,783 training clips that contain the audio (out of 7,000 proposed by [MZA⁺19]) to finetune model on the MSR-VTT.

Zero-shot Step Action Localization. We further evaluate our model on zero-shot step action localization tasks on two datasets: CrossTask [ZAC⁺19b] and Mining YouTube [KIRG19]. The CrossTask dataset consist of 2.7k instructional videos over 18 different tasks. The Mining YouTube dataset provides 250 testing cooking video equipped with an ordered list of action steps. To perform step localization, we use a sliding window and compute the similarity between the current video segment and all step names of the task. Following the inference procedure in [ZAC⁺19b], we obtain the final labeling by running dynamic programming to find the best labeling with respect to the given order of steps based on similarities. We report average recall over all tasks as defined in [ZAC⁺19b]. For both datasets, we use a 3-second sliding window with 1-second step and predict the action for the central time-stamp using a fused va representation.

Method	Train. Mod.	Retrieval	Train. Dataset	Visual BB	Trainable BB	YouCook2				MSR-VTT			
						t	v	a	$R@1 \uparrow$	$R@5 \uparrow$	$R@10 \uparrow$	MedR $\downarrow$	$R@1 \uparrow$
ActBERT [ZY20]	tv	$t \rightarrow v$	HT100M	Res3D+Faster R-CNN	$\checkmark$	9.6	26.7	38.0	19	8.6	23.4	33.1	36
Support Set [PHA ⁺ 21]	tv	$t \rightarrow v$	HT100M	R152 + R(2+1)D-34		-	-	-	-	8.7	23.0	31.1	31
HT100M [MZA ⁺ 19]	tv	$t \rightarrow v$	HT100M	R152 + RX101		6.1	17.3	24.8	46	7.5	21.2	29.6	38
NoiseEstim. [ABARB20]	tv	$t \rightarrow v$	HT100M	R152 + RX101		-	-	-	-	8.4	22.0	30.4	36
Ours	tv	$t \rightarrow v$	HT100M	R152 + RX101		11.2	28.5	39.7	19	9.6	26.1	36.1	23
Ours	tva	$t \rightarrow v$	HT100M	R152 + RX101	$\checkmark$	10.7	27.9	38.9	19	10.3	24.6	35.3	25
MMT [GSAS20]	tva	$t \rightarrow va$	HT100M	7 experts	$\checkmark$	-	-	-	-	-	14.4	-	66
AVLNet [RBH ⁺ 21]	tva	$t \rightarrow v + a$	HT100M	R152+RX101	$\checkmark$	19.9	36.1	44.3	16	8.3	19.2	27.4	47
MCN[CRD ⁺ 21]	tva	$t \rightarrow v + a$	HT100M	R152+RX101	$\checkmark$	18.1	35.5	45.2	-	10.5	25.2	33.8	-
Ours	tva	$t \rightarrow va$	HT100M	R152+RX101	$\checkmark$	20.0	40.7	51.3	10	8.9	23.8	31.8	30
Models with a stronger visual backbone:													
MMV [ARS ⁺ 20]	tva	$t \rightarrow v$	HT100M+AudioSet	TSM-50x2	$\checkmark$ $\checkmark$ $\checkmark$	11.7	33.4	45.4	13	9.3	23.0	31.1	38
VATT [CRD ⁺ 21]	tva	$t \rightarrow v$	AudioSet	Transformer	$\checkmark$ $\checkmark$ $\checkmark$	-	-	45.5	13	-	-	29.7	49
MIL-NCE [MAS ⁺ 20]	tv	$t \rightarrow v$	HT100M	S3D	$\checkmark$	15.1	38.0	51.2	10	9.9	24.0	32.4	29.5
Ours	tva	$t \rightarrow v$	HT100M	S3D†	$\checkmark$	19.8	42.9	55.1	8	9.9	24.0	32.6	28
Ours	tva	$t \rightarrow va$	HT100M	S3D†	$\checkmark$	24.6	48.3	60.4	6	9.3	22.9	31.2	35
FrozenInTime [BNVZ21a]	tv	$t \rightarrow v$	CC+WV+COCO	Transformer	$\checkmark$ $\checkmark$	-	-	-	-	24.7	46.9	57.2	7
CLIP4Clip [ABARB20]	tv	$t \rightarrow v$	WiT + HT100M	CLIP	$\checkmark$ $\checkmark$	-	-	-	-	32.0	57.0	66.9	4

Table 4.1: Zero-shot text-to-video retrieval results on YouCook2/MSR-VTT. In “Retrieval” column: $v + a$ stands for averaging video and audio embeddings for a video representation, va - our joint video-audio embedding where modalities attend to each other during embedding computation, t and v are single-modality embeddings. S3D† is the S3D pretrained by MIL-NCE [MAS⁺20]. We include CLIP4CLIP and FrozenInTime for completeness, but do directly compare because of different pretraining setups. Train Mod.=Training Modalities, BB=Backbone, CC=Conceptual Captions [SDGS18], WV=WedVid-2M [BNVZ21a].

4.4.3 Comparison to State-of-the-Art

Zero-shot Text-to-video Retrieval. First, we assess the performance of the learned multi-modal representation in context of zero-shot text-to-video retrieval task on YouCook2 and MSR-VTT datasets as shown in Table 4.1. In case of the YouCook2 dataset, our method achieves state-of-the-art results over all baselines, including methods with trainable visual and text backbones or a stronger visual backbone as well as methods that do not train the visual backbone. Particularly, our approach improves the AVLnet [RBH⁺21] and MCN [CRD⁺21] baselines that use the same visual, text, and audio backbone and also train with three modalities by increasing $R@10$ from 45.2% to 51.3%. For MSR-VTT however, it shows that a fusion of video and audio modalities is not so beneficial and best performance is reached when considering only text to video retrieval and leaving out audio information. We attribute this behaviour to the domain shift between HowTo100M and MSR-VTT datasets as audio of the HowTo100M dataset mainly contains speech and text as a transcription of speech, while in MSR-VTT, audio can be much less related to the textual description. This assumption is supported by the fact that best-performing methods on MSR-VTT do not use HowTo100M for training at all, such as FrozenInTime [BNVZ21a] or CLIP4CLIP [LJZ⁺21]. Notably, we can further strengthen our model on YouCook2 by leveraging a stronger backbone such as S3D [XSH⁺18], pretrained on HowTo100M by MIL-NCE [MAS⁺20], reaching $R@10$ over 60%. Again, these results show that this better adaptation to HowTo100M does not necessarily translate to better results on MSR-VTT. In Appendix B.1 we additionally include experiments with a stronger CLIP [RKH⁺21c] backbone.

Text-to-video Retrieval after Finetuning. We further evaluated the retrieval performance after finetuning on downstream tasks in Table 4.2. Note that, since several experimental splits were proposed for the MSR-VTT dataset, we report only baselines that used the same training split

Method	Train. Mod.	Retrieval	Pretrain. Dataset	Visual BB	Trainable BB t v a	YouCook2				MSR-VTT			
						R@1 $\uparrow$	R@5 $\uparrow$	R@10 $\uparrow$	MedR $\downarrow$	R@1 $\uparrow$	R@5 $\uparrow$	R@10 $\uparrow$	MedR $\downarrow$
ActBERT [ZY20]	tv	$t \rightarrow v$	HT100M	Res3D+Faster R-CNN		-	-	-	-	16.3	42.8	56.9	10
HT100M [MZA ⁺ 19]	tv	$t \rightarrow v$	HT100M	R152 + RX101		8.2	24.5	35.3	24	14.9	40.2	52.8	9
NoiseEstim. [ABARB20]	tv	$t \rightarrow v$	HT100M	R152 + RX101		-	-	-	-	17.4	41.6	53.6	8
Ours	tv	$t \rightarrow v$	HT100M	R152 + RX101		13.7	35.3	48.4	12	21.0	49.3	60.1	5
Ours	tva	$t \rightarrow v$	HT100M	R152 + RX101	$\checkmark$	12.7	33.9	45.8	13	20.4	47.7	59.3	6
AVLNet [RBH ⁺ 21]	tva	$t \rightarrow v + a$	HT100M	R152 + RX101	$\checkmark$	30.2	55.5	66.5	4	22.5	50.5	64.1	5
MCN [CRD ⁺ 21]	tva	$t \rightarrow v + a$	HT100M	R152 + RX101	$\checkmark$	28.2	53.0	63.7	5	-	-	-	-
Ours	tva	$t \rightarrow va$	HT100M	R152 + RX101	$\checkmark$	32.1	59.1	70.9	3	23.7	52.1	63.7	4

Table 4.2: Text-to-video retrieval on YouCook2/MSR-VTT in the finetune setting. In “Retrieval” column: $v + a$ stands for averaging video and audio embeddings for a video representation, tva - our joint video-audio embedding where modalities attend each other during embedding computation, t and v are single-modality embeddings. Train Mod.=Training Modalities, BB=Backbone.

Method	Tr. Mod.	Tr. BB v	Visual Backbone	Recall $\uparrow$	
				CrossTask	MYT
CrossTask [ZAC ⁺ 19b]	tv		R152 + I3D	31.6	-
HT100M [MZA ⁺ 19]	tv		R152 + RX101	33.6	15.0
MIL-NCE [MAS ⁺ 20]	tv	$\checkmark$	I3D	36.4	-
MCN [CRD ⁺ 21]	tva		R152 + RX101	35.1	18.1
Ours	tva		R152 + RX101	39.3	19.4
Models with a stronger visual backbone:					
MIL-NCE [MAS ⁺ 20]	tv	$\checkmark$	S3D	40.5	-
ActBERT [ZY20]	tv		Res3D+Faster R-CNN	41.4	-
UniVL [LJS ⁺ 20a]	tv	$\checkmark$	S3D $\dagger$	42.0	-
Ours	tva		S3D $\dagger$	41.1	19.7

Table 4.3: Zero-shot action localization performance on CrossTask/MiningYouTube(MYT). S3D $\dagger$ is the S3D pretrained by MIL-NCE [MAS⁺20]. Tr Mod=Training Modalities, Tr. BB v =Trainable Backbone for video modality.

as us for a fair comparison. Results demonstrate that the proposed method clearly outperforms prior works on both datasets. Moreover, after finetuning on the MSR-VTT, the model greatly improves performance by utilizing an audio channel.

Zero-shot Step Action Localization. We further evaluate our methods on zero-shot step action localization on the CrossTask and the MiningYouTube (MYT) datasets in Table 4.3. As video representations we again use the fused video and audio modalities. These results show that the proposed approach clearly outperforms the directly comparable MCN approach on both datasets, as well as a fully supervised baseline CrossTask [ZAC⁺19b], HT100M [MZA⁺19] and MIL-NCE [MAS⁺20] with a trainable I3D visual backbone [MAS⁺20]. Moreover, with a stronger S3D backbone our model also gains improvement over MIL-NCE and is comparable to UniVL (with a trainable backbone) and ActBERT (with additional region-based features from Faster-R-CNN).

4.4.4 Ablation Study

Impact of fusion components. We first address the question of how the proposed components: transformer layer, transformer fusion, and combinatorial loss, impact the overall performance of our system. To this end, in Table 4.4 we considered the following architectures: 1) *no transformers*:

Configuration	Retrieval	YouCook2		MSR-VTT	
		R@5↑	R@10↑	R@5↑	R@10↑
1) no transformers	$t \rightarrow v + a$	32.7	41.4	24.1	33.7
2) single-mod. transformer per mod.	$t \rightarrow v + a$	39.9	50.7	25.3	33.9
3) fusion transformer	$t \rightarrow v + a$	39.5	50.2	23.8	32.7
4) fusion transformer	$t \rightarrow va$	36.6	47.0	22.6	32.1
5) fusion transf. + comb. loss	$t \rightarrow v + a$	38.2	49.2	23.3	33.2
6) fusion transf. + comb. loss (ours)	$t \rightarrow va$	40.7	51.3	23.8	31.8

Table 4.4: Evaluation of the contribution of the proposed fusion transformer and the combinatorial loss. In “Retrieval” column: $v + a$ stands for independently extracting video and audio embeddings and summing up both outputs, while va for forwarding both modalities together allowing them to attend to each other.

Configuration	YouCook2		MSR-VTT	
	R@5↑	R@10↑	R@5↑	R@10↑
ours	40.7	51.3	23.8	31.8
ours + shared final proj.	37.2	48.7	23.1	29.4
ours + $[cls]$ token + shared final proj.	35.9	46.9	21.4	28.8

Table 4.5: Evaluation of different design choices for fusion transformer (with/without $[cls]$ token) and final projection (shared/modality specific) into multi-modal embedding space.

our architecture without transformer and with three pairwise contrastive losses; 2) *single modality transformer*: using three separate modality-specific transformer layers to learn three projection functions; 3) *fusion transformer*: using the proposed modality agnostic transformer but trained with three pairwise contrastive losses without fused modality components; 4) *fusion transformer + comb. loss*: using the proposed modality agnostic transformer with combinatorial input, trained with combinatorial loss to obtain the proposed method. Schematic visualization of these four setups is included in Appendix B.3. We further consider two ways to forward two modalities, first by forwarding them separately and summing up both outputs ($v + a$) and, second, by forwarding them together (va). Overall we observe that adding a simple transformer to process each modality separately already significantly improves performance compared to the baseline, especially for the YouCook2 dataset. We further observe that the overall performance depends on the combination of model, loss function and fusion strategy at test time. While token fusion with transformers is overall beneficial, the best performance is achieved in the *fusion transformer + comb. loss* setup. When using *fusion transformer* alone, the performance drops a bit compared to the *single modality transformer*. However, utilizing shared transformer with combinatorial loss to fuse tokens from different modalities outperforms modality *summing* in the *single modality transformer* setup.

Token Aggregation and Projection. We further evaluate the impact of separate processing of output tokens compared to aggregating information in the $[cls]$ token. To this end, we compare our model architecture to a setup with a shared final projection, as well as a setup with an additional $[cls]$ input token (similarly to BERT [DCLT18]). In the last scenario, we use the output $[cls]$ of the multi-modal fusion transformer as an aggregated representation of input tokens and apply the final shared projection to map it into the shared embedding space. As shown in Table 4.5, our modality-specific projections with no $[cls]$ token benefit over others options on both datasets.

Positional Embedding and Testing on Longer Clips. Finally, we address the question how the option of having random length inputs impacts the overall performance of the model

Configuration	YouCook2		MSR-VTT	
	R@5↑	R@10↑	R@5↑	R@10↑
positional emb. + uniform sampling	32.7	43.4	21.8	29.1
positional emb. + max polling	33.2	43.9	21.6	30.5
positional emb. + averaging over clips	35.6	46.9	23.0	29.2
no positional emb. + uniform sampling	34.5	45.1	22.1	30.2
no positional emb. + max polling	34.8	45.2	23.0	31.8
no positional emb. + averaging over clips	36.9	47.9	22.7	31.3
no positional emb. + video at once	40.7	51.3	23.8	31.8

Table 4.6: Evaluation of different strategies to process long videos (longer than clips used in the training), as well as impact of positional embeddings to encode positional information.

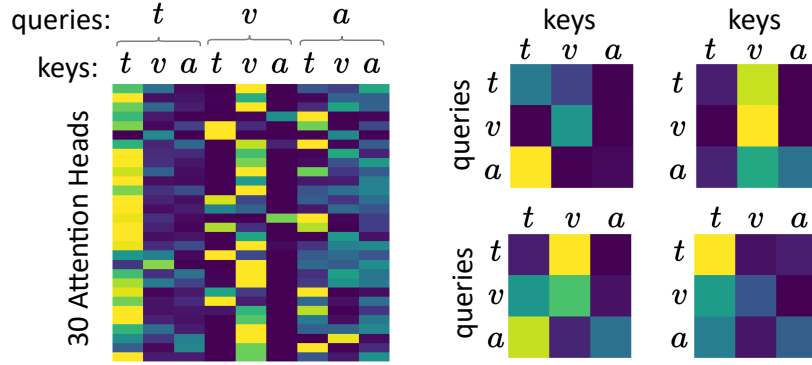


Figure 4.3: Attention Analysis. Left: Average attentions of 30 heads over 128 video clips for queries and keys from different modalities. Right: average attention for random 4 heads presented in 3x3 matrix.

at test time. We consider four different scenarios for testing on longer clips as shown in Table 4.6: 1) *uniform sampling* - after obtaining initial local features from backbones, features are uniformly sampled to fit the maximum number of tokens; 2) *max-pooling* - local features are merged via adaptive max-pooling; 3) *averaging over clips* comprises slicing a longer clip into train-time-length clips and averaging obtained representations; 4) *video at once* considers processing all features at once as proposed. We further compare the first three settings in a scenario with positional embedding to no positional embedding. As positional embedding, we used vanilla trainable embeddings [DCLT18] that are summed up to the input tokens before being input to the transformer. It shows that setups with positional embedding perform almost consistently below the setups without positional embedding on both datasets. Looking at results of different processing strategies, we find that our model benefits from leveraging local temporal dependencies in data via slicing a video clip into shorter clips or *video at once* compared to *max-pooling*, *uniform sampling*. Moreover, utilizing all input data at *video at once* further significantly boosts performance.

Attention Analysis. Finally, we qualitatively analyze the fusion capability of our multi-modal transformer. In Figure 4.3 we show the average attention for query-key pairs of tokens from different modalities. We observe that some heads have a strong attention for single-modality fusion, mostly for *t* and *v* modalities, and in between, some heads are responsible for cross-modal attention.

4.5 CONCLUSION

In this chapter, we propose a multi-modal, modality agnostic fusion transformer approach that learns to exchange information between multiple modalities, such as video, audio, and text, and to integrate them into a joined multi-modal representation, while training solely on freely available instructional videos without relying on human-annotated data. We show that training the system with a combinatorial loss on any possible combinations of modalities allows the fusion transformer to learn a strong multi-modal embedding space leaving out any add-ons such as position encoding.

A clear limitation of the system becomes evident when comparing performance on two downstream datasets, YouCook2 and MSR-VTT: while stronger fusion improves results on some benchmarks, it can reduce generalizability to multi-modal data collected in different ways. This effect arises from the strong correlation between audio and text in instructional videos used in training, where text is typically obtained from audio using automatic speech recognition. As a result, the learned representation tends to reflect this shortcut, focusing primarily on spoken words whenever audio is present, for example, in video–audio embeddings. Later in Chapter 7, we address this issue by improving the quality of text annotations in instructional videos in an unsupervised way, showing that this leads to a significant performance boost of our fusion transformer. And in Chapter 6, we show how our fusion transformer can be further finetuned on downstream domains (e.g., YouCook2 and MSR-VTT) by relying only on the language used in those domains, thereby bridging the language gap between instructional videos and target datasets. Additionally, in Chapter 5, instead of relying on multimodal training with video data to learn strong video representations, we explore how to adapt image–language models for video understanding without the need for supervised finetuning.

In this part of the thesis, Part I, we explore learning strong, generalizable representations from annotation-free data that can either enable zero-shot capabilities for diverse downstream tasks or serve as a strong basis for transfer. In particular, in Chapter 3, we investigate training image representations from unlabeled image data for image understanding, and in Chapter 4, we study learning fused video–audio–text representations for video understanding. In the next part, Part II, we then turn to the problem of adapting our strong, generalizable representations to particular tasks or domains – again without relying on human annotations.

II

FINETUNING MULTIMODAL MODELS WITHOUT HUMAN SUPERVISION

While in the previous chapter we explored methods for pretraining generalized representations from unlabeled data, in this chapter we focus on finetuning them for specific tasks or domains without relying on annotated data in that domain.

In Chapter 5, we demonstrate how to finetune image–language foundation models, which are typically biased toward object recognition, for video understanding, namely action recognition, with an emphasis on activities, using only unlabeled videos and action dictionaries, without requiring any paired human annotations.

In Chapter 7, we take another step by exploring how to finetune vision–language representations for a particular video domain (e.g., cooking), focusing on text-to-video retrieval task, not only without any paired annotations but even without target-domain video data, relying solely on target-domain language.

MATCH, EXPAND AND IMPROVE: UNSUPERVISED FINETUNING FOR ZERO-SHOT ACTION RECOGNITION WITH LANGUAGE KNOWLEDGE

Contents

5.1	Introduction	60
5.2	Related Work	61
5.2.1	Vision-language Models	61
5.2.2	Zero-Shot Action Recognition	62
5.3	Method	62
5.3.1	CLIP on Video Data	63
5.3.2	Text Bag Construction	64
5.3.3	Multiple Instance Learning	65
5.4	Experimental Evaluation	65
5.4.1	Datasets	65
5.4.2	Implementation Details	66
5.4.3	Zero-Shot Action Recognition	66
5.4.4	Few-Shot Action Recognition	67
5.4.5	Ablation Study	68
5.5	Conclusion	71

Large-scale image–language models have shown tremendous success in aligning representations between visual and textual modalities. Compared to video–language models, current image–language models are trained on significantly larger amounts of data, resulting in stronger generalization ability and broader vocabulary coverage. This enables remarkable progress in zero-shot recognition, image generation & editing, and many other exciting tasks, including video applications, such as action recognition. However, image–language models tend to over-represent objects while paying much less attention to verbs, and require additional tuning on video data for best zero-shot action recognition performance. While previous work relied on large-scale, fully-annotated data to finetune an image–language model for action recognition, in this work, we propose an unsupervised approach. We adapt an image–language model for zero-shot and few-shot action recognition using a collection of unlabeled videos and an unpaired action dictionary. Based on that, we leverage Large Language Models and image–language models to build a text bag for each unlabeled video via matching, text expansion and captioning. We use those bags in a Multiple Instance Learning setup to adapt an image–text backbone to video data. Although finetuned on unlabeled video data, our resulting models demonstrate high transferability to numerous unseen zero-shot downstream tasks, improving the base image–language model performance by up to 14%, and even comparing favorably to fully-supervised baselines in both zero-shot and few-shot video recognition transfer.

This chapter is based on the ICCV 2023 paper [LKS⁺23]. Nina Shvetsova contributed to the project through regular scientific discussions.

5.1 INTRODUCTION

Image–language models, often referred to as vision–language (VL) models [RKH⁺21a, LLXH22a, JYX⁺21b] have met unprecedented success in unlocking many vision applications [RKH⁺21a] to work with potentially unlimited open vocabularies, through the promise of zero-shot transfer [ZFZ⁺22, ZYLL22a, ZYLL22b, GLKC22, RMK⁺22, ZGJ⁺22, LWB⁺22, RZC⁺22]. This is empowered by the alignment between visual and language representation spaces, which is effectively attained by VL models leveraging huge amounts of paired image and text data. Incorporating a VL model as a source (base) model or as an architectural component has allowed scaling finetuning on relatively small datasets (e.g. limited in terms of the number of observed objects or other visual concepts compared to the vast VL pretraining) towards zero-shot transfer at inference time. Such zero-shot transfer includes recognizing [ZFZ⁺22, ZYLL22a, ZYLL22b], detecting [GLKC22, RMK⁺22, ZGJ⁺22], segmenting [LWB⁺22, RZC⁺22], and even generating [RLJ⁺22] objects unseen during the finetuning stage and only encountered for the first time at the inference stage.

However, despite the progress in zero-shot image tasks, VL models have been observed to underperform when applied to zero-shot action recognition on video data without any finetuning [WXL21, NPC⁺22, JHZ⁺22, WSO23, CH22, RKM⁺22]. A possible reason, as extensively studied in several works [TJB⁺22, ZZZ⁺22, YBK⁺23a, HN21], is that VL models have a tendency to mostly represent objects (nouns) and not actions (verbs or verb phrases). Therefore, to deal with these shortcomings of VL models w.r.t. zero-shot action recognition, previous works [WXL21, NPC⁺22, JHZ⁺22, WSO23, CH22, RKM⁺22] have used datasets with full annotation (e.g. K400 [KCS⁺17]) to finetune VL models (e.g. the most popular CLIP [RKH⁺21a]) towards improved video zero-shot recognition performance. The potential downsides of this approach are: (i) reliance on full annotation of large-scale action datasets that is time-consuming and cost-intensive, and (ii) the exposure of the model to only the limited action vocabulary during the supervised finetuning (e.g. 400 actions of K400 vs. over 8K possible single verb actions and much more possible general actions in English language) limiting the performance of zero-shot transfer to unseen action categories. In this context, we propose ‘MATCH, eXpand and Improve’ (MAXI) – to allow finetuning on completely unlabeled video data (e.g. unlabeled K400 [KCS⁺17]) and a set of language sources, such as unpaired action dictionaries, Large Language Models (LLM) (e.g. GPT-3 [BMR⁺20]), and VL models for matching (e.g. CLIP [RKH⁺21a]) and captioning (e.g. BLIP [LLXH22a]). To this end, MAXI relies on individual bags of potential texts, collected and refined based on the different language sources, that correspond to each video in the unlabeled set. It then applies Multiple Instance Learning (MIL) for finetuning the VL model using those bags as illustrated in Figure 5.1. We extensively evaluate MAXI on seven downstream zero-shot and few-shot transfer action recognition benchmarks completely unseen during training. We show that MAXI is effective in leveraging unlabeled video data, not only significantly (up to 14%) improving the source VL model performance on all of those tasks, but also favorably competing with state-of-the-art supervised methods trained on fully supervised counterparts of the same finetuning data, and even improving upon them in some zero-shot and few-shot action recognition transfer tasks.

We summarize the contributions of this work as follows.

- We propose MAXI, an approach that leverages an unlabeled video collection and a set of language sources to improve downstream zero-shot action recognition.
- We propose to match each unlabeled video with *text bags* of knowledge mined from the language sources, and employ Multiple Instance Learning for finetuning a VL model

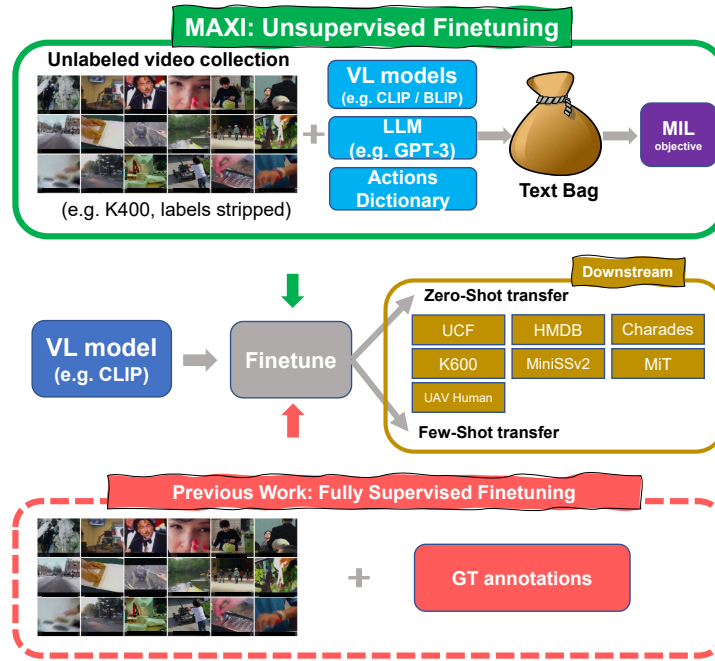


Figure 5.1: While previous work relied on full annotation of action datasets which is time-consuming and cost-intensive to collect, our approach MAXI finetunes the VL model with unlabeled video data. Specifically, we leverage a set of language sources (action dictionary, VL model and LLM) to construct a text bag for each unlabeled video, and employ the Multiple Instance Learning (MIL) objective for finetuning. MAXI demonstrates outstanding improvement of zero-shot and few-shot transfer on downstream novel action datasets.

using these text bags.

- We extensively evaluate our approach on seven unseen action recognition benchmarks, and demonstrate up to 14% absolute zero-shot performance improvements over the source VL model, and even outperform baseline models trained in a fully supervised manner on the same data.

5.2 RELATED WORK

5.2.1 Vision-language Models

Vision-language (VL) Models revolution started with CLIP [RKH⁺21a] and ALIGN [JYX⁺21b] which demonstrated that very large scale (in hundreds of millions) pretraining, on a dataset with massive amount of noisy image-text pairs collected from the web, leads to significant advances in many diverse downstream zero-shot tasks. VL models optimize for image-text alignment via contrastive learning objectives. Earlier methods, such as [TB19, CLY⁺20, LYL⁺20], relied on pretrained object detectors to extract region features. To relax this limitation, cross-attention layers with self-supervised learning objectives, image-text matching, and masked/auto-regressive language modeling were proposed in [KSK21, JYX⁺21b, YDT⁺22, LLXH22a]. BLIP [LLXH22a] combined several techniques for multi-task VL pretraining, achieving strong results in several downstream VL tasks, such as image retrieval, visual question answer-

ing (VQA), image captioning, and reasoning tasks. Finer-level text-image alignment was attempted in [GBB⁺22, YHH⁺21, FRT⁺21, LLZ⁺21d, GLX⁺22], employing additional losses and logic on top of the base contrastive loss of CLIP. FILIP focuses on fine-grained contrastive learning, maximizing the token-wise similarity between image and text tokens. CyClip [GBB⁺22] employs geometrical consistency between the image and text embeddings. DeCLIP [LLZ⁺21d] retrieves nearest neighbors for expanding the set of positive contrastive matches. While these methods have strong zero-shot results on many image benchmarks, such as ImageNet [RDS⁺15] and MS-COCO [LMB⁺14], recent studies such as VL-CheckList [ZZZ⁺22], the Winoground Challenge [TJB⁺22] and ARO [YBK⁺23a], show that these models cannot well distinguish fine-grained language details or understand more structured concepts such as actions that commonly require understanding temporal concepts, movement, and relations between objects. In this work, we show how VL models can be adapted to better understand actions given unlabeled video data.

5.2.2 Zero-Shot Action Recognition

Zero-shot action recognition is the task of recognizing actions that have not been seen during training. This requires the bridging between visual features and semantic representations. Previous works use manually defined attributes [LKS11, ZC17], and word embeddings of action names [BTZ⁺20, MND⁺19, QLS⁺17, SQL20] or action descriptions [CH21a, QYLH22, WC17] as the semantic representation. ER-ZSAR [CH21a] and JigsawNet [QYLH22] leverage crawled descriptions of action classes with manual correction, which require efforts of human annotators for modifying the descriptions. The class descriptions are assigned to the videos based on ground truth labels. On the contrary, our text bag construction requires neither manual correction efforts nor ground truth annotation of videos.

Recent work contributes to adapting large-scale VL model for video understanding, including zero-shot action recognition tasks [WXL21, NPC⁺22, JHZ⁺22, WSO23, CH22, RKM⁺22]. ActionCLIP [WXL21], Ju et al. [JHZ⁺22] and XCLIP [NPC⁺22] adapt CLIP for video data with additional components for spatio-temporal modeling, and demonstrate performance improvements on video tasks. The most recent ViFi-CLIP [RKM⁺22] shows that frame-level processing with feature pooling achieves better visual-language alignment, and outperforms sophisticated related approaches with additional learnable spatio-temporal components. In this work, we follow the architecture and finetuning paradigm of ViFi-CLIP.

Despite the various contributions in architecture design and optimization, the related approaches still rely on ground truth annotations in finetuning CLIP for zero-shot action recognition tasks. Furthermore, no additional language source other than simple action names is explored during finetuning. MAXI overcomes these two limitations by finetuning CLIP (1) without any ground truth labels, and (2) expanding action names by LLM text expansion and visual captioning.

5.3 METHOD

In this work, we propose an approach that effectively leverages a collection of unlabeled videos and a predefined action dictionary (a potentially noisy collection of possible action text labels) to finetune the CLIP model without any ground truth annotations. The purpose of finetuning is to adapt CLIP to video data and to facilitate subsequent Zero-Shot (ZS) transfer to video recognition tasks on novel video categories which are not seen during training. We denote the

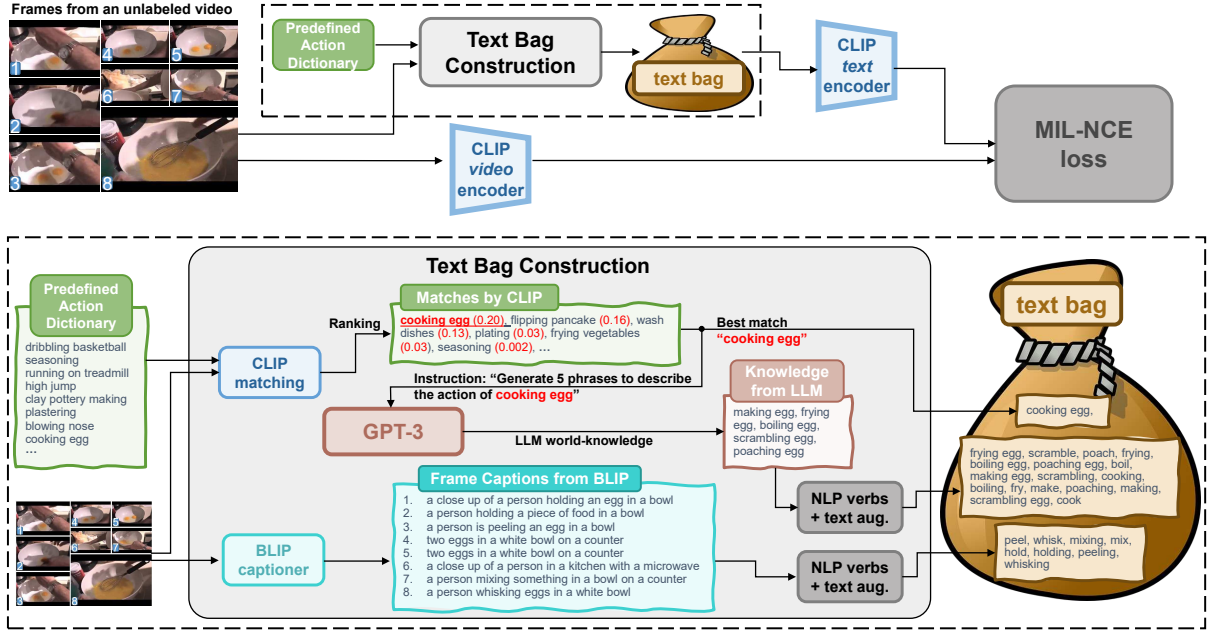


Figure 5.2: Pipeline of MAXI. Given an unlabeled video collection and a predefined action dictionary, we construct a text bag for each video. We finetune CLIP by passing the video and text bag through the adapted CLIP video encoder (Section 5.3.1) and CLIP text encoder, and optimizing with the Multiple-Instance Learning objective (Section 5.3.3). The text bag construction (Section 5.3.2) for an unlabeled video consists of (1) CLIP matching (2) GPT-3 text expansion and (3) BLIP captioning for video to text expansion.

predefined action dictionary as D , and the unlabeled video collection as $V = \{x_j | j \in I\}$, with an index set $I = \{1, \dots, N_V\}$.

Our pipeline is illustrated in Figure 5.2. We first adapt the CLIP image encoder to a video encoder for deployment on video data (Section 5.3.1). Second, given the unlabeled video collection V and a predefined action dictionary D , we use different language sources to construct a *text bag* for each video (Section 5.3.2). The text bag is a (noisy) collection of texts that potentially correspond to the video contents. Third, we perform Multiple Instance Learning (MIL) to learn from the unlabeled videos and noisy text bags (Section 5.3.3), which allows to robustly finetune CLIP in an unsupervised manner.

5.3.1 CLIP on Video Data

CLIP [RKH⁺21a] consists of a visual encoder $\phi_v(\cdot; \theta_v)$ and a text encoder $\phi_t(\cdot; \theta_t)$. We aim to adapt the CLIP image encoder for processing videos. It is demonstrated in [RKM⁺22] that frame-level processing on CLIP image encoder with feature pooling helps in implicitly modeling the temporal cues. This also leads to improved performance over related approaches that additionally incorporate learnable spatio-temporal components. Therefore, following [RKM⁺22], given a video x , we pass M frames into the visual encoder and compute the average of frame features as the video representation, i.e. $z_v = \sum_m \phi_v(x_m^F; \theta_v) / M$. An advantage of this paradigm is that the network can be initialized directly from a large-scale pretrained VL model (e.g. CLIP pretrained on 400M web image-text pairs [RKH⁺21a]) without adding any randomly initialized parameters. This provides a good starting point with reasonable initial performance

before finetuning. We also explore extending a non-randomly-initialized-parameters paradigm to include, e.g., a parameter-free temporal-aware module (see Appendix C.3.4), confirming [RKM⁺22] that a sophisticated temporal module does not lead to better video adaptation from CLIP.

During inference, given a set of class prompts $C = \{t_c\}_{c=1}^{N_c}$, the text feature is computed as $z_{t_c} = \phi_t(t_c; \theta_t)$. For simplicity, we denote the L2-normalized video feature and text feature as $z_v = \bar{\phi}_v(x)$ and $z_t = \bar{\phi}_t(t)$. The zero-shot classification is performed by selecting the class prompt with the maximum similarity to the video representation, i.e., $\hat{c} = \arg \max_c \bar{\phi}_v(x)^\top \bar{\phi}_t(t_c)$.

5.3.2 Text Bag Construction

Given an unlabeled video collection V and a predefined action dictionary D (where each item is a short sentence or a verb phrase describing an action, see Figure 5.2), we construct a text bag T_i for each video $x_i \in V$, i.e. a noisy collection of text prompts describing the video contents.

Predefined action dictionary. In a practical scenario, we usually expect to have coarse prior knowledge of the potential action types in an unannotated video collection. The prior knowledge defines the action dictionary. To have a reasonable action dictionary, we include category names of the action dataset we use for finetuning CLIP. However, the prior knowledge we could obtain in a practical case might not be completely accurate. Therefore, we also explore two cases of noisy action dictionary: a) an under-specified dictionary comprised of only part of possible actions in the set, and b) an over-specified dictionary - adding noisy verbs and verb phrases randomly collected from another text corpus. An evaluation of these settings is given in Section 5.4.5.2.

CLIP matching. For a video $x_i \in V$, we use the original CLIP to match x_i with texts in D w.r.t the cosine similarity. We denote the Top-1 matched text as

$$\hat{t}_i = \arg \max_{t \in D} \text{sim}(\phi_v(x_i), \phi_t(t)) \quad (5.1)$$

where $\text{sim}(u, v) = u^\top v / (\|u\| \|v\|)$ is the cosine similarity. We include $\hat{t}_i$ in the text bag T_i .

The CLIP matching is a means of distilling knowledge from the original CLIP as the teacher. Common choices of unlabeled video collection V are usually of much smaller scale than the original CLIP domain and might be prone to overfitting. Using knowledge from the original CLIP prevents the model from overfitting to the smaller domain V , preserving the generalizability learned in the pretraining stage of CLIP. This hypothesis is supported by experiments in Section 5.4.3 and Section 5.4.4, where we show that compared to all supervised finetuning baselines, the proposed unsupervised pretraining significantly improves zero-shot transfer as well as few-shot adaptation to other novel datasets.

GPT-3 text expansion. We expand the text bag by leveraging the large-scale language model (LLM) GPT-3 [BMR⁺20]. We build upon the fact that GPT-3 has high performance on language instruction tasks [BMR⁺20]. By providing the best-matched text $\hat{t}_i$ in the instruction for LLM requiring it to describe this text using its language (world) knowledge (see instruction example in Figure 5.2), we obtain a collection of expanded alternative descriptions of the action. The descriptions contain details hallucinated by the LLM leveraging its collective world knowledge. We collect the verbs and verb phrases extracted from the generated expanded action descriptions. Furthermore, we perform text augmentation by including both the lemma and gerund (present participle) forms of the verbs. We add the collection of words to the text bag T_i .

BLIP captioning for video to text expansion. We employ the vision-language model BLIP [LLXH22a] for generating captions of individual frames on a video. Note that this image captioning model is not pretrained on any video domain. The frame captions provide instance-level descriptions that are dependent on the visual content of frames of the unlabeled videos. Similar to the case of GPT-3 text expansion, we collect verbs and verb phrases from these descriptions, and perform text augmentation (as stated above), adding the resulting texts to the text bag T_i .

Filtering text bags. To improve the quality of the text bags, we set a threshold δ_p on the similarity score from CLIP matching. We determine δ_p such that $p \times 100\%$ of videos (or text bags) remain after thresholding. For video $x_i \in V$, we keep the corresponding text bag T_i if the best matched text $\hat{t}_i$ has a similarity above the threshold, i.e. $\text{sim}(\phi_v(x_i), \phi_t(\hat{t}_i)) \geq \delta_p$. The filtering results in a sampled index set $I_p = \{i \mid \text{sim}(\phi_v(x_i), \phi_t(\hat{t}_i)) \geq \delta_p, \forall i \in I\}$ and video set $V_p = \{x_i \mid i \in I_p\}$.

5.3.3 Multiple Instance Learning

We employ Multiple Instance Learning (MIL) to learn from the unlabeled videos and noisy text bags collected above. The MIL-NCE loss proposed in [MAS⁺20] combines Multiple Instance Learning and Noise Contrastive Estimation. Following MIL-NCE, instead of enforcing the match of one specific positive text to each video, we softly associate a text bag T_i with each video $x_i \in V$, in which one or multiple texts could be a positive match to the video. As different videos have varying numbers of texts in bag, we randomly sample N_{bag} texts from the original bag in each training iteration. We refine the definition of the sampled text bag T_i as $T_i = \{t_{i,n}\}_{n=1}^{N_{\text{bag}}}$, where N_{bag} is the constant bag size.

The original MIL-NCE loss encourages the instance-level match between each video and its corresponding text bag. In this work, we further propose to encourage the videos and text bags, which have the same best matched text, to be close to each other. Noting that each video x_i has a best matched text $\hat{t}_i$ in the dictionary from CLIP matching step, than our proposed loss is

$$\mathcal{L} = -\frac{1}{|I_B|} \sum_i \log \frac{\sum_j \sum_n \exp(\bar{\phi}_v(x_i)^\top \bar{\phi}_t(t_{j,n})/\sigma) \cdot \mathbb{1}(\hat{t}_i = \hat{t}_j)}{\sum_k \sum_n \exp(\bar{\phi}_v(x_i)^\top \bar{\phi}_t(t_{k,n})/\sigma)} \quad (5.2)$$

where $i, j, k \in I_B$ and $n \in \{1, \dots, N_{\text{bag}}\}$. $I_B \subset I_p$ is a sampled batch of indices. $t_{j,n} \in T_j$ is text in a text bag, and σ is a temperature parameter for contrastive learning. $\mathbb{1}(\hat{t}_i = \hat{t}_j)$ is an indicator that x_i and x_j have the same best matched text.

5.4 EXPERIMENTAL EVALUATION

5.4.1 Datasets

We perform the self-supervised finetuning on Kinetics 400 (K400) [KCS⁺17] without any ground truth labels. K400 is the most popular benchmark for action recognition tasks, containing around 240K training videos for 400 classes. We evaluate action recognition zero-shot transfer and few-shot transfer on several benchmark datasets: UCF101 [SZS12], HMDB51 [KJG⁺11], MiniSSv2 [CPR⁺21] (subset of SSv2 [GEKM⁺17a]), Kinetics600 (K600) [CNBH⁺18], Charades [SVW⁺16], UAV Human (UAV) [LLZ⁺21b], and Moments-in-Time (MiT) [MAZ⁺19]. UCF, HMDB and K600 are collections of online user videos, which are closer in terms of style

Method	GT	Language	Vis.encoder	Frames	UCF101	HMDB51	K600 Top1	K600 Top5
ER-ZSAR [CH21a]	yes	Manual description	TSM	16	51.8 $\pm$ 2.9	35.3 $\pm$ 4.6	42.1 $\pm$ 1.4	73.1 $\pm$ 0.3
JigsawNet [QYLH22]	yes	Manual description	R(2+1)D	16	56.0 $\pm$ 3.1	38.7 $\pm$ 3.7	-	-
ActionCLIP [WXL21]	yes	K400 dict.	ViT-B/16	32	58.3 $\pm$ 3.4	40.8 $\pm$ 5.4	66.7 $\pm$ 1.1	91.6 $\pm$ 0.3
XCLIP [NPC ⁺ 22]	yes	K400 dict.	ViT-B/16	32	72.0 $\pm$ 2.3	44.6 $\pm$ 5.2	65.2 $\pm$ 0.4	86.1 $\pm$ 0.8
A5 [JHZ ⁺ 22]	yes	K400 dict.	ViT-B/16	32	69.3 $\pm$ 4.2	44.3 $\pm$ 2.2	55.8 $\pm$ 0.7	81.4 $\pm$ 0.3
ViFi-CLIP [RKM ⁺ 22]*	yes	K400 dict.	ViT-B/16	16	74.9 $\pm$ 0.6	50.9 $\pm$ 0.7	67.7 $\pm$ 1.1	90.8 $\pm$ 0.3
ViFi-CLIP [RKM ⁺ 22]	yes	K400 dict.	ViT-B/16	32	76.8 $\pm$ 0.7	51.3 $\pm$ 0.6	71.2 $\pm$ 1.0	92.2 $\pm$ 0.3
Text4Vis [WSO23]	yes	K400 dict.	ViT-L/14	16	-	-	68.9 $\pm$ 1.0	-
CLIP [RKH ⁺ 21a]	no	-	ViT-B/16	16	69.9 $\pm$ 1.3	38.0 $\pm$ 1.7	63.5 $\pm$ 0.4	86.8 $\pm$ 0.4
MAXI	no	K400 dict.	ViT-B/16	16	76.6 $\pm$ 0.9	50.5 $\pm$ 0.9	70.4 $\pm$ 0.8	91.5 $\pm$ 0.3
MAXI	no	K400 dict, GPT ₃ verbs	ViT-B/16	16	77.8 $\pm$ 0.3	51.6 $\pm$ 0.9	71.6 $\pm$ 1.0	92.3 $\pm$ 0.3
MAXI	no	K400 dict, GPT ₃ verbs	ViT-B/16	16/32	77.8 $\pm$ 0.5	51.9 $\pm$ 1.1	71.6 $\pm$ 1.0	92.4 $\pm$ 0.3
MAXI	no	K400 dict, GPT ₃ verbs, BLIP verbs	ViT-B/16	16	78.2 $\pm$ 0.8	52.2 $\pm$ 0.6	71.4 $\pm$ 0.9	92.5 $\pm$ 0.3
MAXI	no	K400 dict, GPT ₃ verbs, BLIP verbs	ViT-B/16	16/32	78.2 $\pm$ 0.8	52.3 $\pm$ 0.7	71.5 $\pm$ 0.8	92.5 $\pm$ 0.4

Table 5.1: Zero-shot action recognition on UCF101, HMDB51 and K600. We report mean and standard deviation of results on three official validation splits. All models (except for the original CLIP) are trained on K400. We set the text bag filtering ratio p to 90%. We train with 16 frames per video and report single-view inference results with 16 and 32 frames here. *denotes our re-evaluation.

to K400. The remaining datasets cover larger domain shifts to K400, varying from egocentric motions (MiniSSv2), human and animal videos (MiT), drone videos with small subject in frame (UAV) and 30-second long-term home videos (Charades). More details about datasets are given in Appendix C.1.

We follow the evaluation protocol of zero-shot and few-shot action recognition from [RKM⁺22, NPC⁺22]. We report mAP for multi-label classification on Charades and Top1/Top5 accuracy for single-label classification on the remaining datasets.

5.4.2 Implementation Details

We employ CLIP with the ViT-B/16 [DBK⁺21a] visual encoder. We follow the full-finetuning configuration of [RKM⁺22] to finetune both the visual and text encoder. We consistently set the temperature σ to 0.02. For zero-shot setting, we finetune on K400 without any ground truth labels. We use the AdamW optimizer [LH19] with an initial learning rate of 5×10^{-6} and a cosine decay scheduler. We sample 16 frames from each video and train with a batch size of 256 for 10 epochs. For few-shot learning, we sample 32 frames per video. We set the learning rate to 2×10^{-6} , and train with a batch size of 64 for 50 epochs. During inference, we sample 1 view from each video. Inspired by [WIK⁺22, IWG⁺22], we perform linear weight-space ensembling between the original CLIP (with ratio of 0.2) and the finetuned model. In the main results, we set the text bag filtering ratio p to 90% and bag size to 16.

5.4.3 Zero-Shot Action Recognition

We finetune CLIP on the large-scale K400 dataset stripped of the original ground truth labels. We perform zero-shot action recognition on seven different datasets to verify that cross-dataset model generalizability transfer after the finetuning. In zero-shot setting, the model is evaluated directly on downstream datasets with unseen classes, without being trained on any samples of these datasets.

Method	GT	Language	Charades	MiT	MiniSSv2	UAV
ViFi-CLIP [RKM ⁺ 22]	yes	K400 dict.	25.77	21.68 / 44.19	5.98 / 19.04	4.67 / 15.18
CLIP [RKH ⁺ 21a]	no	-	19.80	20.11 / 40.81	3.96 / 14.42	1.79 / 7.05
MAXI	no	K400 dict.	23.47	21.94 / 45.68	5.19 / 17.71	2.42 / 8.39
MAXI	no	K400 dict., GPT3 verbs	23.74	<u>22.11</u> / <u>45.79</u>	5.60 / 16.73	<u>2.77</u> / <u>9.07</u>
MAXI	no	K400 dict., GPT3 verbs, BLIP verb	<u>23.79</u>	22.91 / 46.38	6.37 / <u>18.73</u>	2.72 / 9.00

Table 5.2: Zero-shot action recognition on Charades, MiT, MiniSSv2 and UAV. All models (except for CLIP) are trained on K400. We report the mAP of multi-label classification on Charades and Top-1/Top-5 single-label classification accuracy for MiT, MiniSSv2 and UAV. We set the text bag filtering ratio p to 90%.

In Table 5.1, we first compare to other state-of-the-art methods, all of which use K400 to adapt CLIP models for zero-shot recognition tasks on UCF, HMDB and K600. Following [RKM⁺22, NPC⁺22, CH21a], we report the mean and standard deviation of results on three official validation sets. ER-ZSAR [CH21a] and JigsawNet [QYLH22] are zero-shot action recognition approaches that train with K400 ground truth annotations. They leverage crawled descriptions of action classes with manual correction, which requires efforts from human annotators. Afterwards, the class descriptions are assigned to videos based on ground truth annotations. We see that the original CLIP has good direct zero-shot performance across the three datasets, which performs better or on par with ER-ZSAR [CH21a] and JigsawNet [QYLH22]. The rest of the compared approaches all adapt CLIP models on video-text pairs with the K400 ground truth class labels as texts. Among them, the most recent ViFi-CLIP [RKM⁺22] achieves the best result, outperforming all the other approaches, without adding any learnable spatio-temporal modules (as done by other approaches such as [WXL21, JHZ⁺22, NPC⁺22]).

In a similar full finetuning paradigm to ViFi-CLIP, MAXI achieves favorable results without using any ground truth annotation. We report the performance of MAXI with different combinations of language sources. Simply with the original K400 action dictionary, we already outperform most of the related work across the three datasets. With the additional GPT-3 verbs and BLIP verbs in the text bag, we further boost the performance, achieving the state-of-the-art among the three datasets.

For a thorough analysis of the model generalizability, we further report the performance of MAXI on four datasets (Charades, MiT, MiniSSv2 and UAV) with larger domain shift to K400 in Table 5.2. In comparison to the original CLIP, our finetuned model has improved zero-shot transfer on all datasets. With the additional language sources of GPT-3 and BLIP, we even outperform ViFi-CLIP trained with ground truth of K400, on the challenging MiT and MiniSSv2 datasets.

5.4.4 Few-Shot Action Recognition

We perform few-shot all-way action recognition to evaluate the model learning capacity in a low data regime. In this setting, we specifically verify whether our self-supervised finetuning on K400 provides a proper initialization for few-shot learning. We follow the few-shot configuration of ViFi-CLIP [RKM⁺22] and XCLIP [NPC⁺22], and use the same training samples in 2, 4, 8 and 16-shot experiments without additional language source for a fair comparison. We train with 32 frames per video. We use the best backbone of self-supervised finetuning (from Section 5.4.3) as the model initialization for few-shot training. In Table 5.3, we report few-shot results of MAXI on three datasets, and also the zero-shot performance of

Dataset Shots	Pretrain on K400	Sett.	HMDB51				UCF101				SSv2			
			2	4	8	16	2	4	8	16	2	4	8	16
CLIP [RKH ⁺ 21a]	no	ZS	41.9	41.9	41.9	41.9	63.6	63.6	63.6	63.6	2.7	2.7	2.7	2.7
ActionCLIP [WXL21]	no	FS	47.5	57.9	57.3	59.1	70.6	71.5	73.0	91.4	4.1	5.8	8.4	<u>11.1</u>
XCLIP [NPC ⁺ 22]	no	FS	53.0	57.3	62.8	64.0	48.5	75.6	83.7	91.4	3.9	4.5	6.8	10.0
A5 [JHZ ⁺ 22]	no	FS	39.7	50.7	56.0	62.4	71.4	79.9	85.7	89.9	4.4	5.1	6.1	9.7
ViFi-CLIP [RKM ⁺ 22]	no	FS	<u>57.2</u>	62.7	<u>64.5</u>	66.8	80.7	85.1	90.0	92.7	6.2	7.4	8.5	12.4
MAXI	yes w/o gt	ZS	49.2	49.2	49.2	49.2	77.8	77.8	77.8	77.8	4.8	4.8	4.8	4.8
ViFi-CLIP [RKM ⁺ 22]	yes gt	FS	55.8	<u>60.5</u>	64.3	65.4	84.0	86.5	<u>90.3</u>	<u>92.8</u>	<u>6.6</u>	<u>6.8</u>	<u>8.6</u>	11.0
MAXI	yes w/o gt	FS	58.0	60.1	65.0	<u>66.5</u>	86.8	89.3	92.4	93.5	7.1	8.4	9.3	12.4

Table 5.3: Few-shot action recognition on HMDB, UCF and SSv2. We report few-shot learning results with and without pretraining on K400.

Matching	Ratio p	Matching acc. on K400	UCF101	HMDB51	K600	MiniSSv2	Charades	UAV	Human	Moments-in-time
CLIP [RKH ⁺ 21a] (w/o finetune)	Zero-Shot		69.93	38.02	63.48	3.96	19.80	1.79		20.11
gt	100%	100%	82.39	52.68	73.39	5.61	25.31	4.47		23.79
CLIP matching	100%	59.7%	77.88	51.09	71.24	5.46	23.52	2.53		22.44
CLIP matching	90%	64.3%	78.17	52.24	<u>71.43</u>	6.37	23.79	2.72		<u>22.91</u>
CLIP matching	50%	80.9%	78.18	50.35	70.78	<u>5.74</u>	<u>23.89</u>	<u>3.06</u>		22.41
CLIP matching	30%	89.5%	76.71	47.73	70.57	4.92	23.14	2.89		21.96

Table 5.4: Text bag filtering with different filtering ratio p . We report the CLIP matching accuracy (after filtering) on K400, and the zero-shot transfer performance of models finetuned with the filtered K400 videos and text bags.

our initialization as a reference. We compare with related approaches that directly perform few-shot learning on CLIP. For a fair comparison, we include the result of few-shot training with a CLIP model that is pretrained with ground truth labels in the ViFi-CLIP paradigm.

We see that few-shot learning using a MAXI-pretrained backbone leads to best performance in most settings, even outperforming the fully-supervised pretrained backbone of ViFi-CLIP. The performance gap is significant in the more challenging extremely limited data scenarios (e.g. 2-shot on HMDB and UCF). Pretraining with full supervision as an initialization might lead to degraded performance in the following few-shot learning (e.g. 8-shot on HMDB, 4-shot on UCF), while our self-supervised finetuned model mitigates this problem, indicating improved generalizability.

5.4.5 Ablation Study

5.4.5.1 Text bag filtering

To improve the quality of text bags used in training, we set a threshold δ_p on the similarity score from CLIP matching, such that $p \times 100\%$ of videos with highest similarity scores remain after the thresholding (see Section 5.3.2). We perform CLIP matching between unlabeled K400 videos and the K400 action dictionary, and use the filtered videos and text bags for finetuning CLIP. In Table 5.4, we report the matching accuracy (after filtering), and zero-shot transfer performance of models finetuned with the filtered K400 videos and text bags. As a reference, we also report CLIP zero-shot performance, and the case of finetuning on 100% accurate video-textbag pairs using ground truth annotation, which leads to the best zero-shot transfer on most datasets.

In Table 5.4, we notice that the CLIP matching accuracy increases continuously with

Action dictionary	Dictionary size	UCF101	HMDB51	K600	MiniSSv2	Charades	UAV	Human	Moments-in-time
CLIP [RKH ⁺ 21a] (w/o finetune)	Zero-Shot	69.93 / 92.7	38.02 / 66.34	63.48 / 86.80	3.96 / 14.42	19.80	1.79 / 7.05	20.11 / 40.81	
K400	400	78.18 / 96.03	50.35 / 77.10	70.78 / 92.17	5.74 / 17.70	23.89	3.06 / 9.46	22.41 / 45.83	
MiniKinetics	200	75.10 / 95.82	<u>48.34</u> / <u>76.95</u>	69.23 / 90.92	6.50 / 18.76	<u>22.70</u>	<u>2.40</u> / <u>8.04</u>	22.50 / 46.01	
K400+WebVid2.5M	800	<u>75.99</u> / <u>96.00</u>	45.97 / 73.94	69.14 / <u>91.13</u>	4.81 / 15.79	<u>22.67</u>	2.11 / 8.00	20.92 / 43.99	

Table 5.5: Robustness of finetuning with noisy action dictionaries. We report the zero-shot transfer performance (mAP on Charades and Top1/Top5 accuracy on other datasets). We set the text bag filtering ratio $p = 50\%$ for improved text bag quality.

decreasing filtering ratio p . Setting $p = 90\%$ leads to consistent improvement of zero-shot transfer, in comparison to the case of $p = 100\%$ due to improved quality of matched texts. Setting $p = 50\%$ leads to partial improvement compared to $p = 100\%$. Further reducing p to 50% leads to performance degradation due to the limited amount of data. This indicates that selecting text bags that CLIP is confident about ensures improved finetuning for more effective zero-shot transfer. However, there is a trade-off between the quality of the filtered data and the amount of data used for training.

5.4.5.2 Robustness against noisy action dictionary

In a practical scenario, we have coarse prior knowledge of the potential action types in an unannotated video collection, which defines an action dictionary. However, such knowledge might be noisy. We explore the robustness of our finetuning pipeline against such a noisy action dictionary. We consider two cases of noisy action dictionaries: (1) an under-specified dictionary consisting of only half of the words of the original K400 action dictionary. Specifically, we use the 200 action names from MiniKinetics [CPR⁺21] (a 200-class subset of K400). (2) An over-specified dictionary by adding noisy verbs and verb phrases into the original K400 action dictionary. We parse verbs from the captions in the validation set of the WebVid2.5M dataset [BNVZ21a], and randomly sample 400 verbs to add to the dictionary, resulting in a dictionary of 800 verbs or verb phrases.

In Table 5.5, we report the zero-shot transfer performance of models finetuned with these noisy dictionaries. Here we set the text bag filtering $p = 50\%$ for improved text bag quality. We also report the results with the original K400 action dictionary as a reference. Apparently, using the clean original K400 action dictionary leads to the best zero-shot transfer on most of the downstream datasets. However, using noisy action dictionaries still leads to significant performance boost compared to the CLIP zero-shot results without finetuning. This indicates the robustness of our pipeline with different cases of noisy predefined dictionaries.

5.4.5.3 What words to include in the text bag?

In Table 5.6, we investigate different combinations of words to include in the text bag. Besides the original K400 action dictionary (*K400 dict.*), we explore: (1) *BLIP verbs*: verbs parsed from BLIP captions; (2) *BLIP object nouns*: nouns of objects parsed from BLIP captions; (3) *GPT3 verbs*: verbs and verb phrases from GPT3 text expansion. For a thorough ablation, we set the text bag filtering ratio $p = 100\%$ to keep the full noisy text bag property.

In Table 5.6, we notice that additional language source upon the original K400 action dictionary leads to further improvement in zero-shot transfer. Interestingly, using BLIP verbs has slightly better results than the case of BLIP object nouns. We assume this is because CLIP has a high object bias and is less sensitive to the language of verbs. Finetuning CLIP

Text bag	UCF101	HMDB51	K600
K400 dict.	76.45	47.43	69.98
K400 dict. + BLIP object nouns	76.23	50.15	71.13
K400 dict. + BLIP verbs	76.94	50.92	71.25
K400 dict. + GPT3 verbs	76.98	50.46	71.24
K400 dict. + GPT3 verbs + BLIP verbs	77.88	51.09	71.24

Table 5.6: Combinations of words in text bags. We report the zero-shot transfer performance on UCF, HMDB and K600. For a thorough ablation, we set the text bag filtering ratio $p = 100\%$ to keep the full noisy text bag property.

Objective	UCF101	HMDB51	K600
Cross entropy	74.48	48.69	65.09
NCE	<u>77.26</u>	49.85	70.08
MIL-Max	77.24	49.85	<u>70.71</u>
MIL-NCE only instance-level	76.96	50.48	70.14
MIL-NCE	77.88	51.09	71.24

Table 5.7: Different strategies of learning from text bags. We report the zero-shot transfer performance on UCF, HMDB and K600. For a thorough ablation, we set the text bag filtering ratio $p = 100\%$ to keep the full noisy text bag property.

by injecting verbs leads to better zero-shot performance in action recognition. Consequently, combining BLIP verbs and GPT3 verbs in the text bag leads to the best zero-shot transfer.

5.4.5.4 How to learn from words in text bags?

In Table 5.7, we explore different strategies of learning from words in a text bag: (1) *Cross entropy*: classification in a fixed class space. (2) *NCE*: contrastive learning to encourage instance-level match between a pair of video and text. In this case, we randomly sample one text from the text bag in each iteration. (3) *MIL-Max*: in each iteration, among words in a text bag, we choose the word with the maximum similarity to the video, and pass the similarity in the contrastive loss. (4) *MIL-NCE*: as explained in Section 5.3.3, we softly associate a bag of texts with the video, and sum up the similarities of texts in a bag (5) *MIL-NCE only instance-level*: the *MIL-NCE* on instance-level match between video and text bag, without encouraging videos and text bags with the same best matched text to be close to each other (see Section 5.3.3). In Table 5.7, we see that cross entropy of classification in a fixed class space leads to the most inferior result, while our MIL-NCE achieves the best improvement. Encouraging videos and text bags with the same best matched text to be close to each other also leads to some performance boost in contrast to only instance-level matching.

5.4.5.5 Bag size

We perform an ablation on the bag size in Table 5.8. A bag size of 1 is the same as *NCE* loss with random word sampling in Table 5.7. Increasing the bag size from lower numbers to 8 leads to consistent performance improvements. Using bag size 16 has further slight performance boost. We report our main results with a bag size of 16.

Bag size	UCF101	HMDB51	K600
1	77.26	49.85	70.08
4	77.24	49.84	70.71
8	77.70	50.61	71.35
16	77.88	51.09	71.24

Table 5.8: Effect of bag size. We report the zero-shot transfer performance on UCF, HMDB and K600. For a thorough ablation, we set the text bag filtering ratio $p = 100\%$ to keep the full noisy text bag property.

5.5 CONCLUSION

In this chapter, we consider the task of finetuning strong pretrained image–language models for video understanding, specifically action recognition, in an unsupervised way by leveraging unlabeled video collections together with a set of language sources. To the best of our knowledge, our approach, *MAtch*, *eXpand*, and *Improve* (MAXI), is the first to address this problem. Specifically, we leverage a set of language sources (unpaired action dictionaries, Large Language Models and image-language models) to construct a text bag for each unlabeled video. Then we use the unlabeled videos and text bags to finetune the image-language model with the objective of Multiple Instance Learning. Our extensive evaluation for zero-shot and few-shot action recognition across several unseen action benchmarks demonstrates significant performance improvement over the source image-language model, as well as improvement over baselines trained in a fully supervised manner. In the next chapter, Chapter 6, we also examine the problem of finetuning models without human supervision, focusing on finetuning for the specific text-to-video retrieval task, where the style of the text query plays an important role in downstream performance, moreover, we explore even more challenging scenario where no video collections from the target task are available for finetuning.

IN-STYLE: BRIDGING TEXT AND UNCURATED VIDEOS WITH STYLE TRANSFER FOR TEXT-VIDEO RETRIEVAL

Contents

6.1	Introduction	74
6.2	Related Work	75
6.2.1	Text-Video Retrieval	75
6.2.2	Large-scale Multimodal Pretraining	76
6.3	Task: Uncurated & Unpaired Text-Video Retrieval	77
6.4	Method	77
6.4.1	Pseudo Matching	78
6.4.2	Style Transfer	78
6.4.3	Training and Retrieval	79
6.5	Experimental Evaluation	80
6.5.1	Dataset Details	81
6.5.2	Implementation Details	82
6.5.3	Text Query Style	83
6.5.4	Uncurated & Unpaired Text-Video Retrieval	84
6.5.5	Comparison to State-of-the-Art	85
6.5.6	Efficiency of Style Transfer	85
6.5.7	Ablation Study	86
6.6	Conclusion	87

IN this chapter, we again address the problem of finetuning strong generalizable image-text and video-text models for downstream video tasks without human annotations, focusing in particular on text-to-video retrieval where the style of the text query plays an important role in performance. Large-scale noisy web image-text and video-text datasets have been proven to be efficient for learning foundational vision-language models. However, when transferring them to the task of video retrieval, models still need to be finetuned on hand-curated paired text-video data to adapt to the diverse styles of video descriptions. To address this problem without relying on hand-annotated pairs, we propose a new setting: text-video retrieval with uncurated and unpaired data. In this setup, training utilizes only text queries from the target task, while we assume that uncurated freely-accessible web videos without any paired text-video annotations are also available. To solve this task, we propose an approach, In-Style, that learns the style of the text queries and transfers it to uncurated web videos. Moreover, to improve generalization, we show that one model can be trained with multiple text styles. To this end, we introduce a multi-style contrastive training procedure that improves the generalizability over several datasets simultaneously. We evaluate our model on retrieval performance over multiple datasets to demonstrate the advantages of our style transfer framework on the new task of uncurated & unpaired text-video retrieval and improve state-of-the-art performance on zero-shot text-video retrieval.

This chapter is based on the ICCV 2023 paper [SKSK23]. As the co-first author, Nina Shvetsova led the project jointly with Anna Kukleva, sharing responsibilities equally across all stages, including idea conception, method design, implementation, experiments, and writing.

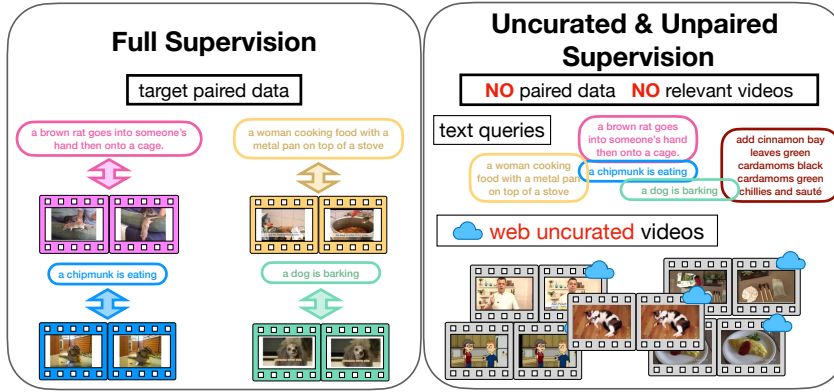


Figure 6.1: Training data for supervised and our uncurated & unpaired settings for text-video retrieval. **Left:** standard supervised text-video retrieval. Aligned and paired data is given for training with the same distribution as the target test set. **Right:** our uncurated & unpaired text-video retrieval. No paired data is available during training, only text queries, whereas to support training, we use uncurated web videos.

6.1 INTRODUCTION

Text-video retrieval refers to the task of retrieving a video from a large pool given a textual description of its content. The field of vision-language retrieval has made remarkable progress, largely driven by models trained on large-scale web collections [RKH⁺21b, LLXH22b, SCR⁺22]. However, while advances in video retrieval often build on pretrained vision-language models to enable better task transfer, most systems still require finetuning on downstream data to adapt the representations to the specific downstream domain. This requires hand-annotated text-video pairs, namely trimmed segments of larger videos that are precisely described by the corresponding texts, for the training and testing of each target downstream dataset. Collecting such aligned pairs of text and videos can be time- and cost-intensive, and particularly gathering videos that comply with national regulations and copyright can be a challenge. Also, in the case of relying on free web content, some videos can become unavailable over time while the respective curated annotations stay available for download but do not have matching videos.

To address this problem, we propose a new setup, *text-video retrieval with uncurated & unpaired data*, assuming the availability of text queries only and without related videos during training (Figure 6.1). The setting is motivated by the fact that it can be considered easier to collect or generate text data, e.g., by producing topic-specific text queries rather than providing a video to match a specific context. To allow the training of a text-video retrieval system based on the given text, we assume to have access to an uncurated video collection as the only source of available videos.

As different domains and datasets contain diverse styles of textual descriptions of videos, we propose a novel method, *In-Style*, to transfer the caption style of given text queries to uncurated web videos, which can be from a deviating distribution compared to the given text queries. To transfer the style of the text queries, we leverage large image-language models [LLXH22b, RKH⁺21b] by creating pseudo pairs that correspond to the given text queries and videos from the uncurated collection by matching them in the shared embedding space [RKH⁺21b]. Thus, we identify a subset of videos that have more similarity to the text queries than the rest of the videos. We then adopt an image-to-text captioning model (a captioner) to mimic the style of our text queries by training with these pseudo pairs. The

stylized captioner is now capable of producing relevant video descriptions in the desired style; therefore, we re-annotate the web videos with the captioner to obtain aligned paired data; we call them generated pairs. Finally, we show that generated pairs help to adapt models pretrained on large-scale web data [LLXH22b, SCR⁺22] to the desired single or multiple styles of given text queries.

We evaluate our model on text-video retrieval over 5 benchmark datasets. Specifically, we demonstrate the advantages of the In-Style method on the new task of uncurated & unpaired text-video retrieval with image-language [LLXH22b] and video-language [SCR⁺22] pretrained backbones. We show the generalization of the proposed approach by training a single model for multiple datasets at once leading to an improved state-of-the-art zero-shot text-video retrieval performance.

We summarize our contributions in the following.

- We introduce a new task of *text-video retrieval with uncurated & unpaired data* where during training, only text queries are available, whereas for the standard text-video retrieval task, paired text-video data is used.
- We propose a novel method, In-Style, to transfer the style of text queries in an unsupervised way, showing that style is an important component for language-based retrieval tasks; therefore, we repurpose large pretrained image-language models to generate pseudo-captions of the same style for uncurated web videos.
- We demonstrate the advantages of our In-Style method for the new task over 5 different datasets with individual models for each dataset as well as one generalized model and we achieve state-of-the-art performance on zero-shot text-video retrieval.

6.2 RELATED WORK

6.2.1 Text-Video Retrieval

Text-video retrieval methods usually focus on learning modules that are able to capture relations between features from text and video modalities [YKK18, LANZ19, GSAS20, CZJW20, CBL⁺21, DKKP21, WZY21b]. Currently, many approaches leverage pretraining on large-scale video-text [BNVZ21a, MZA⁺19, MAS⁺20] or image-text [LLZ⁺21a, LLXH22b] datasets with a further adaptation of the backbone to individually downstream datasets. In this context, ClipBERT [LLZ⁺21a] proposed sparse sampling instead of using dense full-length videos that allow lightweight training. However, foundation models [BHA⁺21] such as CLIP [RKH⁺21b], combine the success of transformer architectures [DBK⁺21b] using a contrastive objective [OLV18] and being trained on large collections of text-image pairs from the web, providing a strong zero-shot [LJZ⁺22, PQOBTM21] baseline on downstream tasks that outperforms many previous methods. Therefore, more recent approaches focus on adapting text-image CLIP pretrained models for text-video retrieval [GVM⁺22, BCJ⁺22, FXXC21, GLC⁺21, LXX⁺22]. X-pool [GVM⁺22] introduces cross-modal attention to reason between text and frames of a video, TS2-Net [LXX⁺22] proposes dynamic adjustments over temporal and spatial token dimensions, which allows finetuning spatial model on video data without architecture changes. Another way to leverage foundation models is to enhance training data [WLF⁺23, ZMKG22]. Cap4Video [WLF⁺23] generates auxiliary captions for available curated training videos by using ZeroCap [TSSW22] that optimizes GPT-2 [RWC⁺19] text generation using a CLIP-based loss [RKH⁺21b]. LaViLa [ZMKG22] proposes to generate additional narrations for a dense

coverage of long videos from the Ego4D dataset [C⁺, GWB⁺22] by finetuning a pretrained large language model [RWC⁺19] on existing annotated text-video paired data. In contrast, we propose to exclude pre-annotated text-video paired data from the training and, relying on text descriptions only, generate text-video pairs leveraging uncurated web videos while transferring the style of original captions.

6.2.2 Large-scale Multimodal Pretraining

Representation learning [RKH⁺21b, LLXH22b, CXH21a, HCX⁺22, ZMKG22, CTM⁺21, YWV⁺22] aims to obtain general representations that improve performance on downstream tasks such as retrieval [LJZ⁺22, PQOBTM21, YWV⁺22, LLXH22b], classification [CXH21a, ZMKG22, CTM⁺21], segmentation [CTM⁺21], question-answering [LLXH22b, YWV⁺22] and captioning [LLXH22b, YWV⁺22]. While some methods rely only on one modality such as images [CTM⁺21, HCX⁺22] or text [RWC⁺19], there is also increasing interest in multi-modal representations [RKH⁺21b, LDF⁺20, SZC⁺20, LCC⁺20, LJS⁺20b, SCR⁺22] which require multi-modal aligned pairs. However, the acquisition of human-annotated paired data is expensive; therefore, noisy web data [RKH⁺21b, MZA⁺19] allows for significant scaling of such datasets. Many methods successfully utilize web image-text pairs [RKH⁺21b, JYX⁺21c, YWV⁺22], whereas uncurated video-text pairs are not only harder to collect but are also more prone to misalignments. Therefore, efforts are made to align ASR (automatic speech recognition) with video frames via contrastive learning [MZA⁺19, MAS⁺20, XGH⁺21, ZLH⁺21a] or in an unsupervised way [HXZ22]. To overcome those issues, we propose to generate synthetic video descriptions with the desired caption style and train models on those captions instead of raw ASR text.

For contrastive-based vision-language representation learning methods, the dual-encoder architecture is a common choice as it features two parallel branches for two modalities, which are contrasted against each other to learn a joint embedding space [RKH⁺21b, SZC⁺20, LCC⁺20, LJS⁺20b]. Recently, BLIP [LLXH22b] and CoCa [YWV⁺22] propose a unified multi-task contrastive-generative framework that combines contrastive and captioning objectives. These methods rely on both curated image-text and uncurated web image datasets, with BLIP additionally iteratively applying the generation and filtering of synthetic captions. Compared to those works, we adopt pretrained image-language models for uncurated & unpaired text-video retrieval by transferring the caption style directly on uncurated videos without any aligned data during training.

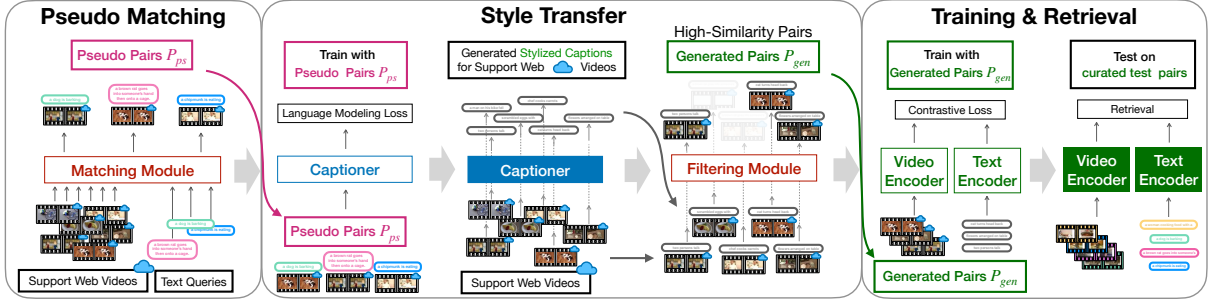


Figure 6.2: **The proposed In-Style method.** First, in the pseudo matching step, pseudo pairs P_{ps} , which consist of text queries and the most related web videos from the support set, are created. Style Transfer: the captioner is tuned with the obtained pseudo pairs P_{ps} to adapt to the style of the given text queries. Next, new stylized captions are generated for all videos in the support set and then filtered to avoid noisy captions; the resulting set of the generated pairs P_{gen} contains web videos and aligned captions of the desired style. To complete the retrieval task, we adapt the dual video-text encoder model with the generated pairs P_{gen} and evaluate it on the curated paired test set.

6.3 TASK: UNCURATED & UNPAIRED TEXT-VIDEO RETRIEVAL

In this section, we introduce the proposed uncurated & unpaired text-video retrieval training setup. Typically, models for text-video retrieval are trained on *paired* text-video data. Given a set of pairs of captions t_i and corresponding videos v_i : $\{(t_i, v_i)\} \in D$, where D is a data distribution, the goal is to learn a similarity function $s(t_i, v_j)$ that calculates the similarity between the caption t_i and the video v_j . The training can be done from scratch, but typically pretrained image-language [RKH⁺21b, LLXH22b] or video-language models [LJS⁺20b, MZA⁺19] are finetuned on the target paired text-video data and then evaluated on the test set from the same distribution D [LXX⁺22, FXXC21]. If the evaluation is performed on multiple datasets, the model is usually finetuned for each dataset individually.

In contrast, we propose *a text-video retrieval with uncurated & unpaired data*, where only target text queries are available during training without any videos. More precisely, given a set of text descriptions $\{t_i\}$ from data distribution D , we aim to learn useful information about the similarity $s(t_i, v_j)$ in D relying only on the textual descriptions. We further assume that a large set of freely accessible web videos $V' = \{v'_j\} \in D'$ without any paired text is available to support the training (such as videos of the HowTo100M dataset [MZA⁺19]). We note that the data distribution D' in the support video dataset can deviate from the distribution D .

Finally, to avoid training different models individually for each target dataset, we further consider learning a *generalized* model that maintains the performance of individual models over a set of K datasets of different caption styles and coming from different data distributions $D_1, \dots, D_K$.

6.4 METHOD

To address the task of uncurated & unpaired text-video retrieval, we aim to transfer the style of the text queries (the only available curated information) to an uncurated web video dataset. To this end, we rely on web-scale pretrained image-language models as a supervisory signal and leverage them as a matching module and pretrained captioning model that we adapt

throughout the training process. The steps of the proposed In-Style method are shown in Figure 6.2. The first step is *Pseudo Matching*, described in Section 6.4.1, which matches the given text queries to the most relevant videos from the set of all uncurated web videos. The following *Style Transfer* step (Section 6.4.2) adapts the pretrained captioning model (the captioner) to the target text style by training it on the previously obtained pseudo pairs. The captioner is then used to generate new style-adapted captions for all available web videos, which are then filtered to avoid too noisy pairs; we refer to the resulting filtered web videos with style-adapted video descriptions as generated pairs. Finally, we adapt a pretrained vision-language model for the task of text-video retrieval on the generated pairs (Section 6.4.3). Moreover, in Section 6.4.3, we propose the training of a generalized model on multiple styles of text queries at the same time and introduce a new contrastive objective, In-Style, that improves training on more than one text style at once.

6.4.1 Pseudo Matching

First, we obtain pseudo video-text pairs, with each pair containing one of the available text queries and the most relevant uncurated video from the web collection. For pseudo matching, we leverage image-language models such as CLIP [RKH⁺21b] or BLIP [LLXH22b] that excel in zero-shot retrieval performance [LJZ⁺22]. Such models usually follow a dual-encoder architecture: encoders f_t and f_v projects text t and image x into a common multimodal embedding space. The similarity of text and image is computed as a cosine similarity in this common space: $\text{sim}(t, x) = \frac{f_t(t)^\top f_v(x)}{\|f_t(t)\| \|f_v(x)\|}$. We use this metric to match the available text queries to the closest video.

Since available videos can vary in overall duration (for example, five or more minutes) and cover a lot of different actions, we divide all videos into non-overlapping clips of s -seconds. We denote $V' = \{v'_j\}$ as a set of all such video clips. Then, we calculate a multimodal representation for each video clip v'_j as an average representation of m uniformly sampled frames of a video (see Appendix D.2). Using precomputed embeddings, we connect every caption t_i with a video v' with maximum similarity from available set of videos V' , such as:

$$v_i^p = \arg \max_{v'_j \in V'} \text{sim}(t_i, v'_j). \quad (6.1)$$

To increase the diversity of matched videos, we do not allow multiple captions to match the same video clip; therefore, when video clip v_i^p is matched, we exclude it from V' . Thus, we obtain a set of pseudo text-video pairs $P_{ps} = \{(t_i, v_i^p)\}$. In Section 6.5, we show that this step allows us to introduce a weak supervision that may not find the exact match but provides a basis for further style transfer.

6.4.2 Style Transfer

We aim to transfer the style of the given text queries to other unrelated web videos by generating new captions with the desired style. Inspired by the ability of language models conditioned on visual input [LLXH22b] to generate plausible descriptions for diverse visual inputs, we propose to adapt the pretrained image captioner g using the obtained set of noisy pseudo text-video pairs P_{ps} . By doing this, we adapt the captioner to both the style of the captions as well as the style of the web videos. This allows us to generate new stylized captions P_{gen} for the full support set of videos V' using this captioner.

Captioner. More specifically, we follow the BLIP [LLXH22b] captioner architecture, which we extend for video captioning by conditioning the model not only on a single image but on a number of video frames. To this end, we apply the image encoder on each frame individually and inject a joint set of visual tokens into the text decoder model, which produces text in an autoregressive manner. We provide further details in Appendix D.2. To train the captioner g on the pseudo text-video pairs P_{ps} , we utilize the common language model loss that optimizes cross-entropy loss between ground truth and predicted probabilities of the next token given a correct set of previous tokens in the sentence. Following BLIP, we also use label smoothing with parameter 0.1 while calculating cross-entropy.

Stylization of Captions. For each video $v'_i \in V'$, we generate a caption $t_i^s = g(v'_i)$ with a captioner g trained on pseudo pairs by using a nucleus sampling [HBD⁺20]. Nucleus sampling was shown to generate more diverse and detailed captions than a beam search [VCS⁺16, LLXH22b].

Filtering. As the captioner g is adapted on pseudo pairs and shifts the model closer to a vocabulary of given text queries $\mathcal{D}$, some of the generated captions t_i^s might be noisy and not descriptive for the web videos. Therefore, we further filter the generated pairs based on a similarity score $s(t_i^s, v'_i)$ utilizing the large pretrained image-language dual encoders the same way as it was used for creating pseudo text-video pairs (Section 6.4.1). Leaving only pairs with similarity higher a threshold $s(t_j^s, v'_j) > th$, we obtain a paired set of web videos and stylized related captions $P_{gen} = \{(t_j^s, v'_j)\}$. In Section 6.5.7, we show that even a noisy set of pseudo pairs is enough to adapt a captioner for generating captions in a desired text style and that stylized captions combined with the following filtering provide a strong learning signal to boost the performance of retrieval in target distribution D .

6.4.3 Training and Retrieval

Single-Style Training. To allow for text-video retrieval based on the stylized captions and the paired video data, we train a dual-encoder architecture [LLXH22b] on the set of generated pairs P_{gen} with the contrastive loss [OLV18]. We show that P_{gen} provides better supervision than P_{ps} or even a combination $P_{gen} + P_{ps}$. Practically, we consider several pretrained models: the image-text model BLIP [LLXH22b], which we adapted for video as described in Section 6.4.1, as well as video-text model EAO [SCR⁺22], which is pretrained on the HowTo100M dataset with ASR-video pairs, which serve as noisy supervision. Following previous works, we use symmetric contrastive loss, which brings together text t_i^s and video v_i from a text-video pair $(t_i^s, v_i) \in P_{gen}$ (a positive pair) in shared video-text embedding space, and contrasting them on video and text from different pairs (negatives), that are pushed apart:

$$L = -\frac{1}{2B} \sum_{i=1}^B \left(\log \frac{\exp(\frac{s(t_i^s, v'_i)}{\tau})}{\sum_{j=1}^B \exp(\frac{s(t_i^s, v'_j)}{\tau})} + \log \frac{\exp(\frac{s(v'_i, t_i^s)}{\tau})}{\sum_{j=1}^B \exp(\frac{s(v'_j, t_i^s)}{\tau})} \right) \quad (6.2)$$

where τ denotes a temperature parameter and B is a number of pairs.

For the finetuning of the BLIP model, we follow the original setup and utilize the extension of contrastive training with a momentum encoder and a queue that keeps more negatives as well as soft labels. To finetune the EAO model, we follow the respective setup without a momentum encoder or soft labels.

Multi-Style Training. Finally, we consider training a generalized model on multiple sources of text queries coming from different data distributions $D_1, \dots, D_K$. Let's denote $P_{gen}^1, \dots, P_{gen}^N$ set

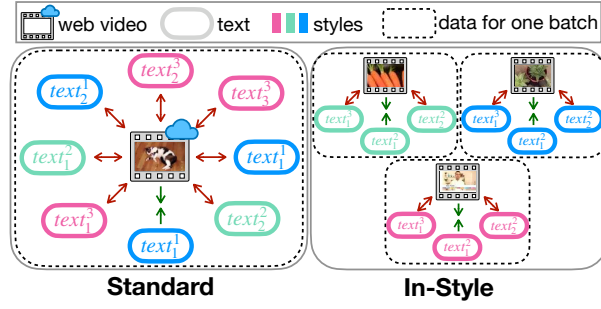


Figure 6.3: **Multi-dataset training.** **Left:** Standard contrastive training with multiple datasets. **Right:** Our In-Style training procedure. Each batch consists of text queries that belong only to the same style. Note that we use only web videos from the support set; therefore, all videos are from the same distribution.

of generated pairs for the captions from $D_1, \dots, D_N$ respectively. Here, different sources can have various styles that might highlight different aspects of videos in their captions (Table 6.3). As an example, captions in the YouCook2 dataset [ZXC18] are more “action”-oriented, e.g., “combine macaroni sauce and cheese” or “stir in crushed tomatos”, while captions of the LSMDC dataset [RRS15] are third-person descriptions, e.g., “Someone gazes at the beautiful animal” or “Someone chews the sweet”. In standard training [LLXH22b], all different styles with their matching videos would be present in contrastive loss together, which can lead to a mixture of different visual topics and text styles, which are easy to separate and which might include only a few hard negatives per sample. To avoid this possibly noisy setting, we propose to modify the training procedure and select video-caption pairs with captions from the same data source for contrastive loss. Formally, during training, we iterate over generated sets of pair $P_{gen}^1, \dots, P_{gen}^N$ sampling a minibatch $\{(t_i^g, v_i')\}_{i=1}^B$ from a single set $P_{gen} \in \{P_{gen}^1, \dots, P_{gen}^N\}$ and calculating loss $L(\{(t_i^g, v_i')\}_{i=1}^B)$ performing one optimization step with a minibatch (Figure 6.3). We note that for BLIP training, we keep separate queues for each set P_{gen} .

We show in Section 6.5.4 that this setting can be beneficial for learning a generalized model. Our intuition is that text queries with the same style provide stronger negatives for the model, allowing the model to concentrate on the content of the captions rather than a style.

6.5 EXPERIMENTAL EVALUATION

We evaluate the proposed uncurated & unpaired text-video retrieval approach on five popular benchmark datasets: MSR-VTT [XMYR16], YouCook2 [ZXC18], MSVD [CD11], LSMDC [RRS15], and DiDeMo [AHWS⁺17]. All datasets cover different styles of captions and videos, which include YouTube and Flickr videos on various topics and video clips from movies. As a source of support videos, we use the large-scale web dataset HowTo100M [MZA⁺19]. We additionally test our model with text queries from the VATEX dataset [WWC⁺19] as well as with third-party text queries (not video captions), specifically with the recipe steps from Food.com dataset [MLNM19] and task descriptions from WikiHow dataset [KW18].

Pretrained Model	Method	Supervision	MSR-VTT				YouCook2				DiDeMo				MSVD				LSMDC				Mean			
			R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR
BLIP [LLXH22b]	Zero-shot	none	34.1	60.2	70.6	3	6.0	16.2	23.1	70	28.2	52.0	62.7	5	38.8	64.8	74.0	2	14.5	29.3	36.4	32.5	24.3	44.5	53.4	22.5
	In-Style (ours)	only text	36.2	61.8	71.9	3	8.6	21.6	30.0	37	32.1	61.9	71.2	3	44.8	72.5	81.2	2	16.1	33.6	39.7	25	27.6	50.3	58.8	14
	GT finetuning	T-V pairs	42.9	69.7	78.9	2	12.6	32.0	43.6	15	40.2	70.6	79.3	2	48.1	76.6	85.0	2	23.8	41.1	50.9	10	33.5	58.0	67.5	6.2
EAO [SCR ⁺ 22]	Zero-shot	none	9.9	24.0	32.6	28	19.8	42.9	55.1	8	6.6	19.0	26.8	42	18.0	40.4	52.3	9	3.6	8.5	13.0	177	11.6	27.0	36.0	52.8
	In-Style (ours)	only text	16.4	35.8	48.9	10	20.3	46.4	58.8	7	13.2	31.6	44	15	23.4	50	62.4	5	4.9	12.3	16.7	94	15.64	35.22	46.16	26.2
	GT finetuning	T-V pairs	22.8	47.8	60.3	6	26.7	55.9	68.6	4	19.2	43.1	54.4	8	25.1	53.6	65.7	5	8.9	21.2	29.4	40	20.5	44.3	55.7	12.6

Table 6.1: **Text-video retrieval with style transfer.** Comparison between the upper bound, where the retrieval model trained with ground truth aligned text-video pairs (T-V pairs), zero-shot respective models (no style transfer or tuning), and our In-Style method, where we follow our new setting of *uncurated & unpaired text-video retrieval* for style transfer based only on input text queries.

Training Dataset	MSR-VTT				YouCook2				DiDeMo				MSVD				LSMDC				Mean			
	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR
MSR-VTT	36.2	61.8	71.9	3	7.6	18.8	25.9	62	29.0	54.5	65.4	4	43.3	70.7	79.9	2	15.2	28.5	35.3	31	26.3	46.9	55.7	20.4
YouCook2	31.5	55.5	64.4	4	8.6	21.6	30.0	37	25.1	53.9	65.2	4	41.1	67.3	76.8	2	14.2	28.8	36.9	30	24.1	45.4	54.7	15.4
Didemo	34.0	58.5	68.9	3	6.8	17.2	24.5	69	32.1	61.9	71.2	3	43.7	71.6	80.5	2	16.6	30.5	38.4	28	26.6	47.9	56.7	21
MSVD	36.0	59.4	69.5	3	6.4	16.4	23.6	70	27.0	54.9	65.0	4	44.8	72.5	81.2	2	14.5	27.4	34.8	32	25.7	46.1	54.8	22.2
LSMDC	33.9	60.3	69.9	3	7.1	18.1	25.6	68	31.7	59.9	69.1	3	44.6	71.7	80.0	2	16.1	33.6	39.7	25	26.6	48.7	56.8	20.2
Target dataset (mean over diagonal)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	27.5	50.3	58.8	14
All five datasets – standard training	36.4	62.1	71.8	3	8.7	21.4	29.4	44	31.4	62.5	71.2	3	44.7	72.9	81.5	2	16.3	31.9	39.5	25	27.5	50.2	58.7	15.4
All five datasets – In-Style (ours)	36.7	61.9	72.3	3	8.5	21.8	30.4	38.5	32.6	61.8	71.2	3	44.7	73.1	82.0	2	16.6	32.2	39.8	26	27.8	50.2	59.1	14.5

Table 6.2: **Generalization performance of different models over all datasets.** Mean denotes an average of R1, R5, R10, MR over 5 datasets, correspondingly. **Top:** the proposed In-Style method with the input text queries only from one **respective** training dataset. **Bottom:** training with 5 different text query styles. Comparison between standard multi-dataset training and proposed In-Style procedure.

6.5.1 Dataset Details

MSR-VTT [XMYR16] contains in total 10k videos on various topics and 200K captions. More precisely, every 20 captions describe the same video in different words. We use split 9K+1K [GSAS20] in evaluation, resulting in 180K captions for training and 1K text-video pairs for testing.

YouCook2 [ZXC18] is a dataset of 13.5K cooking instructional video clips, where each clip is annotated with a short cooking recipe step. Following [MZA⁺19, SCR⁺22], we use a 10K+3.5K training-testing split, leveraging 10K captions for training.

MSVD [CD11] contains 2K video snippets, where each is associated with approximately 40 sentences. The standard split consists of 1200 videos for training, 100 for validation, and 670 for testing. The training set contains 48K captions.

LSMDC [RRS15] is a collection of 202 movies sliced into 118K movie clips with one description per clip and with about 100K clips used for training, and 7408 and 1000 text-video pairs used for validation and testing, respectively.

DiDeMo [AHWS⁺17] is a fine-grained text-video dataset. 10K Flickr videos are paired with multiple detailed sentences (40K sentences in total). During training, we use single sentences (33K captions), whereas for evaluation on the test set, we follow [BNVZ21a] and concatenate all the descriptions for video into one paragraph, acting as a video-paragraph retrieval task (we do not use ground truth time-stamp annotations).

VATEX [WWC⁺19] dataset contains 35K video clips with multiple annotated captions for a video, covering 600 different human activities. The training set contains 260K captions.

Food.com [MLNM19] is a text dataset that contains more than 230K recipe texts with over 2.2M recipe steps crawled from websites. We use recipe steps as text queries in our training.

WikiHow [KW18] is a large-scale text dataset using the online WikiHow knowledge base. The

Dataset	Examples
MSR-VTT (~43 symbols in a text)	1) The peoples are sharing their view on this car of different models 2) Someone is showing the ingredients for a dish they are going to make 3) A man is playing an instrument
YouCook2 (~39 symbols in a text)	1) Combine macaroni sauce and cheese 2) Grate and cube potatoes 3) Stir in crushed tomatos
DiDeMo (~147 symbols in a text)	1) A dog runs down a hill and stop behind a shrub. Dog sniffs and chews at patch of grass on rock. the dog approaches, then begins to sniff the cluster of plants first time hand is seen petting dog. 2) Only big screen is visible the camera first pans to the large screen. The view shifts from the basketball court to the fans in the seats across the stadium. Camera goes to the bigscreens the dancers are shown on the jumbotraun. 3) A bus stops. The bus stops at the end of the driveway. A kid is coming out of a school bus. School bus doors open.
MSVD (~31 symbols in a text)	1) The cats are fighting 2) The lady sliced a vegetable 3) A man is eating a pizza
LSMDC (~46 symbols in a text)	1) SOMEONE goes to the kitchen, wets a towel, comes back to the bed, kneels it, places the towel on SOMEONE's brow. 2) He slaps SOMEONE again. 3) SOMEONE moves off through the crowd.

Table 6.3: Three random examples of text descriptions in different datasets. With the dataset name, we also report the median length of a text in the dataset.

dataset contains more than 230K articles covering a variety of topics/tasks and descriptions of steps to solve these tasks. We use only headline steps as text quires, which gives us 1.7M captions.

HowTo100M [MZA⁺19] is a dataset of instructional videos that cover a large variety of topics. The dataset consists of more than 1M videos that were collected by querying on YouTube 23,000 different “how to” tasks. In our setup, we use 8-second non-overlapping clips from a 100K random subset of videos (no more than 15 clips per video) as a support video dataset, resulting in ~1.4M video clips.

6.5.2 Implementation Details

Model. We leverage the pretrained dual-encoder CLIP (ViT-B/32) model [RKH⁺21b] in the matching module and the filtering module. Captioner weights are initialized with BLIP (ViT-B/16) captioner [LLXH22b], which is pretrained on five different image-text datasets, including LAION [SVB⁺21a] with 129M images. For retrieval, we consider two architectures: dual encoder image-text initialized with BLIP (ViT-B/16) and dual encoder video-text architecture initialized from EAO [SCR⁺22] pretrained on HowTo100M with noisy ASR narrations. We follow [SCR⁺22] and use a model with a S3D [XSH⁺18] feature extractor and weights that were pretrained with a video-text-audio triplet, but only utilize the video-text encoder and report all results without audio.

Method	Image-Text Datasets	Video-Text Datasets	MSR-VTT				YouCook2				DiDeMo				MSVD				LSMDC			
			R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR
HowTo100M [MZA ⁺ 19]	-	HowTo100M	7.5	21.2	29.6	38	6.1	17.3	24.8	46	-	-	-	-	-	-	-	-	-	-	-	-
SupportSet [PHA ⁺ 21]	-	HowTo100M	8.7	23.0	31.1	31	-	-	-	-	-	-	-	-	8.9	26.0	37.9	18	-	-	-	-
VATT [AYQ ⁺ 21]	-	HowTo100M+AS	-	-	29.7	49	-	-	45.5	13	-	-	-	-	-	-	-	-	-	-	-	-
EAO [§] [SCR ⁺ 22]	-	HowTo100M	9.9	24.0	32.6	28	19.8	42.9	55.1	8	6.6	19.0	26.8	42	18.0	40.4	52.3	9	3.6	8.5	13.0	177
Nagrani et al. [NSS ⁺ 22]	-	VideoCC3M	19.4	39.5	50.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Frozen in Time [BNVZ21a]	CC+COCO	WebVid-2M	24.7	46.9	57.2	7	-	-	-	-	21.1	46.0	56.2	7	-	-	-	-	-	-	-	-
CLIP-straight [PQOBT21]	WIT	-	31.2	53.7	64.2	4	-	-	-	-	-	-	-	-	37.0	64.1	73.8	2	11.3	22.7	29.2	56.5
CLIP4CLIP [LJZ ⁺ 22]	WIT	HowTo100M	32.0	57.0	66.9	4	-	-	-	-	-	-	-	-	38.5	66.9	76.8	2	15.1	28.5	36.4	28
Nagrani et al. [NSS ⁺ 22]	WIT	VideoCC3M	33.7	57.9	67.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BLIP [LLXH22b]	CC+COCO+3more [*]	-	33.3	57.3	67.5	3.5	5.8	15.0	21.9	76	24.6	50.4	59.7	5.3	37.0	63.3	72.6	3	15.2	28.2	35.9	35
In-Style (ours) (CLIP)	WIT	HowTo100M [†] +VATEX [‡]	35.0	59.6	70.4	3	5.1	14.0	20.3	103	26.6	50.5	62.6	5	38.6	66.3	77.9	3	16.0	31.6	38.5	26.5
In-Style (ours) (BLIP)	CC+COCO+3more [*]	HowTo100M [†] +VATEX [‡]	36.0	61.9	71.5	3	6.8	16.7	24.5	63	29.4	59.2	68.6	3.5	44.9	72.7	81.1	2	16.4	30.1	38.7	28
In-Style (ours) (BLIP)	CC+COCO+3more [*]	HowTo100M [†] +WikiHow	34.2	59.6	69.0	3	7.3	19.2	27.1	46	29.7	56.2	67.4	4	42.8	70.2	79.1	2	17.0	30.8	39.6	27
In-Style (ours) (BLIP)	CC+COCO+3more [*]	HowTo100M [†] +Food.com	32.8	54.9	65.8	4	7.2	19.8	27.9	47	25.7	52.8	63.1	5	39.5	64.9	74.9	2	14.5	28.9	37.2	30.5
In-Style (ours) (BLIP)	CC+COCO+3more [*]	HowTo100M [†] +Target [†]	36.2	61.8	71.9	3	8.6	21.6	30.0	37	32.1	61.9	71.2	3	44.8	72.5	81.2	2	16.1	33.6	39.7	25
In-Style (ours) (EAO)	-	HowTo100M+Target [†]	16.4	35.8	48.9	10	20.3	46.4	58.8	7	13.2	31.6	44.0	15	23.4	50.0	62.4	5	4.9	12.3	16.7	94

Table 6.4: **Zero-shot comparison with other methods.** **Top:** zero-shot retrieval with methods pretrained on video-language or/and images-language web or/and curated datasets which exclude target datasets during training. For our In-Style method, the VATEX dataset is used as a source of text queries. **Bottom:** uncurated & unpaired text-video retrieval with text queries from the respective target datasets for comparison purposes. Note that this setting is not zero-shot. [†] denotes that only videos were used (without paired text) and [‡] – only text (without videos). [§]For EAO, performance with S3D backbone is reported. ^{||}For BLIP, the performance of dual encoder architecture is reported (not image-grounded text encoder). ^{*}CC [CSDS21]+COCO [LMB⁺14]+VG [KZG⁺17]+SBU [OKB11] +LAION [SVB⁺21a]. AS denotes AudioSet [GEF⁺17].

Training Data	MSR-VTT				YouCook2				DiDeMo				MSVD				LSMDC				Average			
	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR
— (zero-shot)	34.1	60.2	70.6	3	6.0	16.2	23.1	70	28.2	52.0	62.7	5	38.8	64.8	74.0	2	14.5	29.3	36.4	32.5	24.3	44.5	53.3	22.5
Pseudo pairs P_{ps}	35.0	61.4	70.9	3	7.5	19.6	28.9	43	33.1	59.8	71.2	3	44.3	72.4	81.0	2	16.8	32.7	40.4	25	27.3	49.2	58.4	15.2
Generated pairs P_{gen}	36.2	61.8	71.9	3	8.6	21.6	30.0	37	32.1	61.9	71.2	3	44.8	72.5	81.2	2	16.1	33.6	39.7	25	27.6	50.3	58.8	14.0
Combined $P_{ps} + P_{gen}$	36.0	61.3	71.5	3	8.9	21.8	29.8	37	32.6	61.8	70.2	3	44.4	72.2	80.8	2	17.1	32.4	40.4	26	27.8	49.9	58.5	14.2

Table 6.5: **Different types of training pairs for text-video retrieval step.** We evaluate text-video retrieval with pseudo pairs P_{ps} only, with generated pairs P_{gen} only, and the combination of both $P_{ps} + P_{gen}$.

Training. For training, we uniformly sample $m = 8$ frames per video with a resolution of 224×224 augmented with RandAugment [CZSL20]. For captioner training and BLIP-architecture retrieval model training, we use AdamW optimizer [LH19] with a weight decay of 0.05, a batch size of 128, and a learning rate $1.0e - 05$ for captioner and $1.0e - 06$ for retrieval. For the EAO model following [SCR⁺22], we use Adam optimizer [KB15a] without weight decay. More training details can be found in Appendix D.3.

Evaluation. For testing, we use $m = 64$ frames for the fine-grained DiDeMo dataset, and $m = 12$ for all others, following [LJZ⁺22]. For text-video retrieval, we report standard recall metrics for R1, R5, R10, and the median rank (MR).

6.5.3 Text Query Style

We consider text style as a set of attributes and properties of the text shared across a text corpus. Such properties might be the usage of stop words, sentence construction, sentiment, text length, etc. To highlight those differences, we show three text examples from the different datasets in Table 6.3. The respective word clouds for these datasets with and without stop words can be found in Figure D.2 in the Appendix. It shows that the sentence structure and most

Training data	MSR-VTT				YouCook2				DiDeMo				MSVD				LSMDC				Average			
	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR
— (zero-shot)	34.1	60.2	70.6	3	6.0	16.2	23.1	70	28.2	52.0	62.7	5	38.8	64.8	74.0	2	14.5	29.3	36.4	32	24.3	44.5	53.3	22.5
P_{gen} with zero-shot captioner	36.3	61.6	71.8	3	7.1	18.4	25.6	65	28.7	56.3	65.0	4	43.8	71.2	80.1	2	16.0	29.2	37.7	30	26.3	47.3	56.1	20.8
In-Style P_{gen} (non-target)	36.0	61.9	71.5	3	6.8	16.7	24.5	63	29.4	59.2	68.6	3.5	44.9	72.7	81.1	2	16.4	30.1	38.7	28	26.7	48.1	56.9	19.9
In-Style P_{gen} (target)	36.2	61.8	71.9	3	8.6	21.6	30.0	37	32.1	61.9	71.2	3	44.8	72.5	81.2	2	16.1	33.6	39.7	25	27.6	50.3	58.8	14

Table 6.6: **Source of generated pairs P_{gen} for text-video retrieval.** Comparison between zero-shot BLIP (no adaption of retrieval model), zero-shot BLIP captioner, and adapted BLIP captioner with our In-Style method with either text queries from VATEX (non-target) or text queries from the target datasets.

frequent words change across datasets. For example, the YouCook2 test queries always start with an action verb, while in other datasets, the subject+verb+object structure is mostly used. While in the MSR-VTT dataset, frequent words are third-person nouns like “man”, “woman”, “person”, “people”, the DiDeMo uses more words about camera position like “camera”, “left”, “right”, “screen”, “view”, and the LSMDC mostly describes a subject as “someone”. While the MSR-VTT and the MSVD datasets might look similar, Table 6.3 shows that sentences in the MSR-VTT are 1.5 times longer than in the MSVD on average. We consider such properties as style properties of the text.

6.5.4 Uncurated & Unpaired Text-Video Retrieval

Single Dataset Training. First, we demonstrate the efficiency of the proposed style transfer method in uncurated & unpaired text-video retrieval on five different downstream datasets in Table 6.1. We present results for the image-text pretrained BLIP [LLXH22b] model as well as for the video-text pretrained EAO [SCR⁺22] model. We consider three evaluation scenarios: 1) zero-shot performance; 2) the performance of our style transfer method in the text-video retrieval task with uncurated & unpaired data where only text queries are available during training; 3) training with the ground truth aligned text-video pairs, which can be considered as an upper bound for our task. It shows that the proposed In-Style method significantly outperforms zero-shot performance even without using any aligned training samples from the target distribution. This supports the hypothesis that the style of the text queries is an important component of text-video retrieval. Moreover, we observe that the gap between training with ground truth aligned pairs and the style transfer can be remarkably small, especially on the MSVD dataset, indicating the benefits with respect to a potential annotation cost reduction in the proposed setup.

Multi-Dataset Training. Second, we evaluate the proposed multi-dataset training procedure with the In-Style method in Table 6.2. Here, a minibatch is compiled from a single text source as shown in Figure 6.3. This is favorable compared to the standard training, where data points in a minibatch are randomly sampled from all data sources together. It shows that the proposed procedure leads to improved retrieval performance compared to individually trained models and better generalization across all datasets compared to the standard multi-dataset training. We attribute the performance increase compared to standard multi-dataset training to the fact that considering the captions of only the same style in contrastive loss provides a model with a cleaner learning signal with stronger text negative counterparts. As an example, “add sliced cucumber” in YouCook2 style would be a stronger negative in comparison to a correct “add sliced tomato” query than a “a person in a video puts sliced cucumber in a salad” in MSR-VTT style. More discussions of generalization can be found in Appendix D.1.

6.5.5 Comparison to State-of-the-Art

We further compare the proposed method with zero-shot retrieval baselines in Table 6.4. We report the performance of BLIP and CLIP backbones trained with text queries from the VATEX dataset, thus text queries do not follow the distribution of any of the test datasets. The closest counterpart to our model is Nagrani et al. method [NSS⁺22], which utilizes the pretrained image-text CLIP backbone, which is further trained with the VideoCC3M dataset [NSS⁺22] – a video-text dataset collected by automatic transferring image captions from text-image CC3M dataset [SDGS18]. The conceptual difference between [NSS⁺22] and our method is that [NSS⁺22] proposes to transfer *image* captions from the image-caption dataset by pairing images to videos, while the proposed In-Style method adapts the model to the *video* captions. While noting that a direct comparison to different state-of-the-art methods is limited due to different pretraining datasets, it can be observed that the proposed In-Style method achieves the best results on four out of five datasets, underperforming only in YouCook2, which might benefit from HowTo100M pretraining. We additionally validate the statement that text queries can be used without any corresponding videos by using texts from WikiHow [KW18] and Food.com [MLNM19] datasets that contain descriptions of different actions/steps to solve tasks or cook meals. In Table 6.4, we show that style transfer from both datasets especially benefits YouCook2 retrieval performance that we attribute to the similarity in text styles (see Appendix D.1). However, style transfer from the WikiHow dataset, which is more diverse and covers a larger variety of topics, also improves the performance over the baselines on the DiDemo, MSVD, and LSMDC datasets.

6.5.6 Efficiency of Style Transfer

Training Pairs. In Table 6.5, we compare the performance of the models trained either with pseudo pairs P_{ps} or with generated pairs P_{gen} , or with a combination of them $P_{ps} + P_{gen}$. All setups boost the performance of text-video retrieval by a large margin compared to zero-shot text-video retrieval. The generated pairs P_{gen} achieve better performance than pseudo pairs P_{ps} on all datasets except LSMDC, whereas a combination of $P_{ps} + P_{gen}$ does not improve performance on average. We note that the number of pairs in P_{gen} is significantly larger than in P_{ps} (Table 6.7b) for all datasets except LSMDC (a dataset of movies, which might contain a larger domain shift to YouTube videos compared to other datasets). We assume that in this case, P_{gen} contains better-aligned pairs since each generated text description is conditioned on the corresponding video, while in P_{ps} , a fixed set of descriptions is matched (see examples in Figure 6.4) explaining the performance drop with $P_{ps} + P_{gen}$.

Style Transfer. In Table 6.6, we consider how much the text style transfer in the generated pairs P_{gen} influences the retrieval performance. For this, we considered three sets of P_{gen} for the training retrieval model: 1) P_{gen} generated with zero-shot BLIP captioner; 2) In-Style P_{gen} generated with captioner trained on P_{ps} with text queries from a different non-target dataset (we used the VATEX dataset); 3) In-Style P_{gen} with a captioner trained on P_{ps} with text queries from the target dataset. We observe that training the model with generated text-video pairs (from uncurated web videos from the HowTo100M dataset) by a zero-shot image-pretrained captioner already improves the performance in all video retrieval datasets. We attribute this to the content and style adaptation of the image-language model to the specific appearances in the videos. However, such models tend to generate “static” descriptions that do not involve actions. Thus, text queries from non-target video datasets, namely the VATEX dataset, improve the retrieval performance further. Yet, we notice that YouCook2 does not benefit from the VATEX

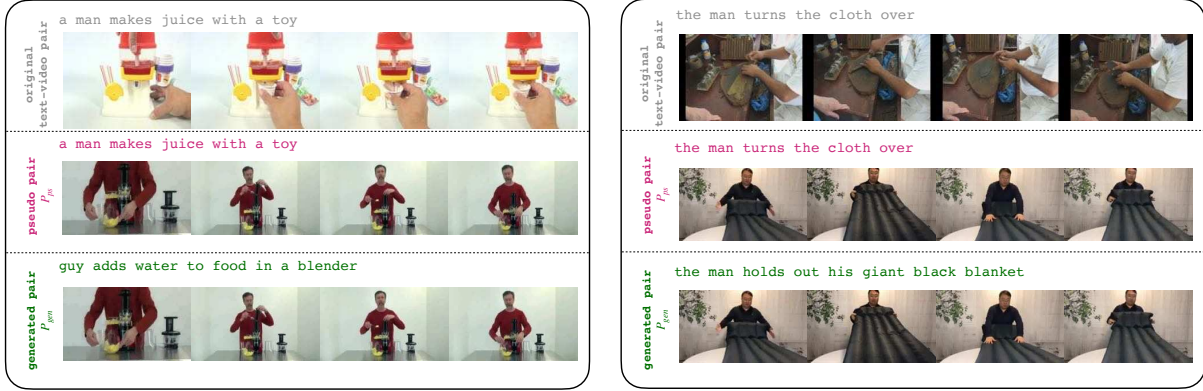


Figure 6.4: **Qualitative evaluation of P_{ps} and P_{gen} on the MSR-VTT (left) and DiDeMo (right) datasets.** First, a text query is matched with one of the videos (a pseudo pair P_{ps}), and then, after the style transfer step, for each video, a new caption is generated in the same style but with updated content (a generated pair P_{gen}).

Filt. Thr.	R1	R5	R10	MR
0.26	43.9	71.8	80.8	2
0.27	44.2	72.2	80.9	2
0.28	44.8	72.5	81.2	2
0.29	45.0	72.3	80.9	2
0.30	45.1	72	80.8	2

(a) Filtering threshold

Matching	R1	R5	R10	MR
Random	39.1	66.3	75.8	2
BLIP	44.1	71.4	80.0	2
CLIP	44.8	72.5	81.2	2

(c) Matching method

Dataset	#Pseudo Pairs	#Generated Pairs
MSR-VTT	180k	495k
YouCook	10k	168k
Didemo	33k	280k
MSVD	48k	379k
LSMDC	101k	144k

(b) Number of P_{ps} and P_{gen}

Training pairs	B@4	ROUGE	CIDEr
– (zero-shot)	0.305	0.519	0.610
Pseudo pairs	0.559	0.628	1.059
GT pairs	0.659	0.680	1.296

(d) Captioning performance

Table 6.7: Ablations of our In-Style method on the MSVD.

text queries as from the zero-shot generated captions. Finally, using training text queries from the target dataset excels on the considered benchmarks.

6.5.7 Ablation Study

Matching Method. To obtain generated pairs, we train the captioner with pseudo pairs that were created by a matching module. In Table 6.7c, we consider two options for the matching module: image-text pretrained dual encoders from BLIP [LLXH22b] and CLIP [RKH⁺21b], as well as the “Random” option where text queries are simply matched with the random videos. We report the text-video retrieval performance of our final model using the given option of the matching module. We observe that the matching module based on CLIP leads to better performance. We attribute that to the robustness of CLIP to the noisy web data as it was trained on large-scale web image-text pairs, whereas BLIP utilizes additional filtering to reduce the noise during training.

Filtering Threshold. In Table 6.7a, we consider the effect of filtering on the quality of the generated pairs P_{gen} . We find threshold $th = 0.28$ works the best, indicating that filtering is an important step for our style transfer framework.

Captioning Performance. Finally, we evaluate the captioning performance of the captioner trained with pseudo pairs P_{ps} with the standard NLP metrics BLEU@4, ROUGE, and CIDEr. Table 6.7d demonstrates that the captioner trained with pseudo pairs almost doubles the zero-shot captioner performance, significantly reducing the gap to the training with ground truth supervision.

6.6 CONCLUSION

In this chapter, we again tackle the problem of finetuning strong generalizable models for downstream tasks without human annotations. Specifically, we address a new task, *text-video retrieval with uncurated and unpaired data*, where only target-domain text queries are available when finetuning a vision-language model for ext-video retrieval in that domain. Motivated by the fact that different domains imply diverse styles of video descriptions, we introduced the In-Style method that preserves the style of the given input queries and transfers it to the support set of unrelated web videos, creating aligned text-video pairs with the style of the input. Utilization of obtained text-video pairs as supervision leads to a significant performance boost in text-video retrieval. Moreover, we show the performance generalization of a single model that we train with multiple styles simultaneously, proposing a training procedure for multi-dataset training. We evaluate the proposed model over multiple datasets and show the advantages of the In-Style method on the task of uncurated & unpaired text-video retrieval and achieve new state-of-the-art results for zero-shot text-video retrieval.

To summarize, in this part of the thesis, Part II, we explored strategies for adapting strong, generalizable representations to particular tasks or domains without relying on annotated data. In Chapter 5, we investigate adapting image-language models for video understanding, focusing on action recognition. In Chapter 6, we study the adaptation of vision-language models for text-video retrieval tasks in different downstream domains.

Together, Part I and Part II cover both pretraining and finetuning strategies for learning strong representations without human annotations across different visual tasks. In the next part, Part III, we turn to the data aspect of building models.

III

IMPROVING DATA QUALITY WITHOUT HUMAN SUPERVISION

While the previous parts covered methods for pretraining and finetuning models with annotation-free data, in this part, we focus on the data side of model development. We introduce strategies to improve the quality of both training and evaluation data, laying the foundation for building more reliable and generalizable vision models.

In particular, in Chapter 7, we propose a framework to generate high-quality textual annotations for large-scale, annotation-free web video data in a training-free and scalable manner, leveraging the strong zero-shot reasoning abilities of LLMs. Using this approach, we construct a large-scale, high-quality text–video–audio dataset that serves as a valuable resource for pretraining video understanding models.

Further, in Chapter 8, we examine the problems of evaluation benchmarks in video understanding and propose a method to detect and mitigate undesirable representation biases in video benchmarks that may unintentionally hinder fair evaluation of video understanding capabilities. As a result, we provide object-debiased test splits for 12 popular video evaluation benchmarks.

HowToCaption: Prompting LLMs to Transform Video Annotations at Scale

Contents

7.1	Introduction	92
7.2	Related Work	93
7.2.1	Large-Scale Video-Language Datasets	93
7.2.2	Learning with Noisy ASR Subtitles of Instructional Videos	94
7.2.3	LLMs in Vision-Language Tasks	94
7.3	Method	95
7.3.1	Problem Statement	95
7.3.2	Video-Language Retrieval Model	95
7.3.3	HowToCaption Method	96
7.3.4	HowToCaption Dataset	97
7.4	Experimental Evaluation	98
7.4.1	Datasets and Metrics	98
7.4.2	Implementation Details	100
7.4.3	Ablation Study	100
7.4.4	Comparison to State-of-the-Art	102
7.5	Conclusion	103

This chapter explores ways to improve large-scale pretraining multimodal datasets collected from the internet without the need for human curation, in particular addressing the challenge of obtaining high-quality dense textual annotations for long-form video-audio data. Instructional videos are a common source for learning text-video or text-video-audio representations by leveraging subtitles extracted with automatic speech recognition systems (ASR) from the audio signal in the videos. However, in contrast to human-annotated captions, subtitles naturally differ from the visual content of the videos and thus provide only noisy supervision. As a result, large-scale annotation-free web video training data remains sub-optimal for training text-video models. In this work, we propose to leverage the capabilities of large language models (LLMs) to obtain high-quality video descriptions aligned with videos at scale. Specifically, we prompt an LLM to create plausible video captions based on ASR subtitles of instructional videos. To this end, we introduce a prompting method that is able to take into account a longer text of subtitles, allowing us to capture the contextual information beyond one single sentence. We further prompt the LLM to generate timestamps for each produced caption based on the timestamps of the subtitles and finally align the generated captions to the video temporally. In this way, we obtain human-style video captions at scale without human supervision. We apply our method to the subtitles of the HowTo100M dataset, creating a new large-scale dataset, HowToCaption. Our evaluation shows that the resulting captions not only significantly improve the performance over many different benchmark datasets for zero-shot text-video retrieval and video captioning, but also lead to a disentangling of textual narration from the audio, boosting the performance in text-video-audio tasks.

This chapter is based on ECCV 2024 paper [SKH⁺24]. As co-first author, Nina Shvetsova jointly led the project with Anna Kukleva during the initial months sharing responsibilities equally across all stages, and, following Anna Kukleva’s shift of focus due to an internship,

Nina Shvetsova continued as the primary driver of the work and lead author. The ideas developed in this work also serve as the foundation for the method presented in the next chapter [SNS⁺25].

7.1 INTRODUCTION

Textual descriptions of visual information allow for navigating large amounts of visual data. Recently, image-text cross-modal learning has achieved remarkable performance in many downstream tasks by pretraining on large-scale web datasets consisting of text-image pairs [RKH⁺21d, SVB⁺21a, JYX⁺21a]. To collect video data on a similar scale, media platforms such as YouTube can be used as a great source of freely available videos [AEHKL⁺16, ZXC18, SLS⁺20, MZA⁺19, ZLH⁺21a, XHZ⁺22]. Most of these videos include some narrations, e.g., in instructional videos [MZA⁺19], people explain and show how to accomplish one or another task. To transform spoken language from the videos into subtitles, current automatic speech recognition (ASR) systems [RKX⁺23] can be used, providing aligned text-video annotated pairs for free. This automatic supervisory signal can easily scale to large video datasets. However, such video web data poses additional challenges [HXZ22, MZA⁺19]: (1) spoken and visual information in the video can deviate from each other, e.g., when speakers provide information beyond what is visible or when spoken instructions do not temporally align with the actions shown, (2) speech contains filler words and phrases, such as “I’m going to”, and can be incomplete and contain grammatical errors, and (3) ASR transcripts usually do not have punctuation and may contain errors (Figure 7.1). Therefore, ASR subtitles provide only weak, noisy supervision for videos.

To address this problem, we propose a new framework, HowToCaption, that leverages large language models (LLMs) [CLL⁺23] to generate human-style captions on a large scale for web-video instructional datasets based on corresponding ASR subtitles (Figure 7.1). By carefully designing prompts, we show that the LLM can effectively map long, noisy subtitles into concise and descriptive human-style video captions. Moreover, we obtain an initial temporal alignment of the generated captions to the video based on the ASR timestamps by tasking the LLM to predict timestamps for each caption. For additional quality improvement, we apply alignment and filtering within short temporal windows with respect to the predicted timestamp. This approach can generate aligned text-video pairs on a large scale without any human intervention.

Beyond providing better annotation, the new captions provide the advantage that they are no longer a direct output of the speech signal, thus effectively decoupling audio and text. Current methods usually avoid using audio [MAS⁺20, HXZ22], as the ASR subtitles are directly derived from the speech, thus leading to the problem that any text-to-audio+video retrieval would mainly retrieve the closest speech signal while disregarding the video [SCR⁺22, NSS⁺22]. Being able to generate captions that deviate from the speech thus allows to extend retrieval to audio+video without the need for finetuned regularization, as used in [SCR⁺22].

To verify the effectiveness of the proposed HowToCaption method, we generate new captions for the large-scale HowTo100M dataset [MZA⁺19], obtaining a new *HowToCaption* dataset. We evaluate the quality of the improved textual descriptions on various challenging downstream tasks over four different datasets, namely YouCook2 [ZXC18], MSR-VTT [XMYR16], MSVD [CD11], and LSMDC [RRS15]. It shows that the generated captions not only provide a better training signal but also allow for a decoupling of speech and caption annotation, allowing a retrieval based on audio, vision, and subtitles at scale. We release the new HowToCaption dataset with high-quality textual descriptions to show the potential of generated captions for

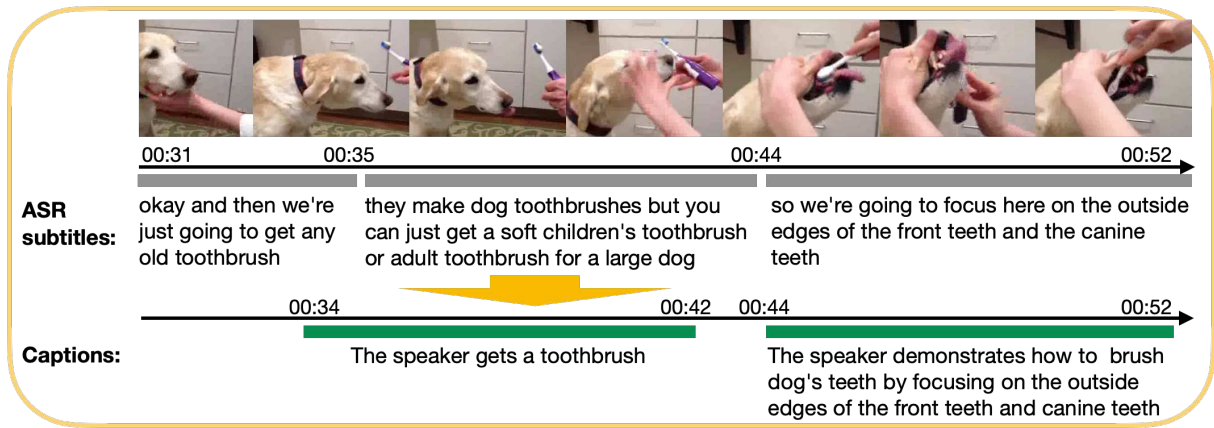


Figure 7.1: ASR subtitles deviate from human-written captions: they contain a lot of filler phrases, e.g., “we’re going to”, and extra information, e.g., “they make dog toothbrushes”. We propose to generate human-style video captions based on the ASR subtitles and their timestamps that we further temporally realign with the video.

web text-video pairs. We summarize the contributions of this work as follows:

- We propose a HowToCaption method to efficiently convert noisy ASR subtitles of instructional videos into accurate video captions, which leverages recent advances in LLMs and generates high-quality video captions at scale without any human supervision.
- We create a new HowToCaption dataset with high-quality human-style textual descriptions with our proposed HowToCaption method.
- Utilizing the HowToCaption dataset for training text-video models allows us to significantly improve the performance over many benchmarks for text-to-video retrieval and video captioning. Moreover, since the new textual annotation allows us to disentangle audio and language modalities in the instructional videos, where the ASR subtitles were highly correlated to audio, we show a boost in text-video+audio retrieval performance.

7.2 RELATED WORK

7.2.1 Large-Scale Video-Language Datasets

Manual annotation of video captioning datasets is extremely time-consuming since it involves video trimming and localization of caption boundaries. Currently, manually annotated datasets, e.g., MSR-VTT [XMYR16], YouCook2 [ZXC18], HIREST [ZCK⁺23], and HT-Step [AMN⁺24], are limited in size. Therefore, different methods of mining videos with weak supervision from the Internet were considered. Datasets such as YouTube-8M [AEHKL⁺16] and IG-Kinetics-65M [GTM19] provided multiple class labels based on query clicks, metadata [AEHKL⁺16] or hashtags [GTM19]. However, short class labels are suboptimal supervision compared to textual descriptions [DJ21]. Therefore, Bain et al. [BNVZ21b] considered scrapping videos with associated alt-text from the web, obtaining the WebVid2M and WebVid10M datasets [BNVZ21b] with 2M and 10M video-text pairs, respectively. Stroud et al. [SLS⁺20] proposed to use meta information, such as titles, descriptions, and tags from YouTube, as a textual annotation and

created the WTS-70M dataset. Nagrani et al. [NSS⁺22] proposed to transfer image captions from an image-text dataset to videos by searching videos with similar frames to an image and collected the VideoCC3M dataset. Yang et al. [YNL⁺24] created the VidChapters-7M dataset by scraping user-annotated chapters on YouTube. However, most videos in WebVid do not have audio, which is an essential part of video analysis, and captions in VideoCC3M are derived from images and, therefore, tend to describe more static scenes rather than actions. At the same time, the title, tags, and chapters of WTS-70M and VidChapters-7M provide only high-level video descriptions.

As an alternative to this, Miech et al. [MZA⁺19] proposed the HowTo100M dataset, where instructional videos are naturally accompanied by dense textual supervision in the form of subtitles obtained from ASR (Automatic Speech Recognition) systems. The HowTo100M dataset with 137M clips sourced from 1.2M YouTube videos was proven to be effective for pretraining video-audio-language representations [RBH⁺21, CRD⁺21, SCR⁺22]. The followed-up YT-Temporal-180M [ZLH⁺21a] and HD-VILA-100M [XHZ⁺22] datasets follow the same idea but contain more videos with higher diversity and higher video resolution. While ASR supervision provides a scalable way to create large datasets with dense annotation, the quality of subtitles is not on par with human-annotated captions. In this work, we propose a method to create high-quality captions for videos at scale by leveraging LLM and subtitles.

Recent works, InternVid [WHL⁺23], VAST [CLW⁺24], and Video-ChatGPT [MRKK23], collect large-scale video-text datasets through per-frame image captioning and text summarization with LLMs. In contrast, our work stands apart in its objective. Rather than distilling existing knowledge from image models, pretrained on human-annotated image-text data, into videos, we aim to gather a dataset enriched with new video knowledge. This is achieved by converting freely available ASR subtitles into captions. Concurrent work, HowToStep [LCH⁺23], summarizes ASR subtitles into descriptive steps using LLM and then performs additional temporal realignment. Our main difference is that we focus on general video captions rather than procedural steps.

7.2.2 Learning with Noisy ASR Subtitles of Instructional Videos

The problem of misalignment and noisiness of ASR supervision in instructional videos, e.g., in the HowTo100M dataset, were addressed in multiple works. MIL-NCE loss [MAS⁺20] and soft max-margin ranking loss [ABARB21] were proposed to adapt contrastive loss to misalignment in text-video pairs. Zellers et al. [ZLH⁺21a] proposed to use LLM to add punctuation and capitalization to ASR subtitles and remove mistranscription errors. Han et al. [HXZ22] proposed to train temporal alignment networks to filter out subtitles that are not alignable to the video and determine alignment for the others. However, to the best of our knowledge, [LPB⁺22] is the only work that goes beyond just removing mistranscription errors and ASR re-alignment, where Lin et al. proposed to match subtitles to step descriptions from WikiHow dataset [KW18] (distant supervision). In our work, we propose to use LLM to create video captions given ASR subtitles, which allows us to create detailed descriptions that are specific for every video and have proper sentence structure.

7.2.3 LLMs in Vision-Language Tasks

In recent years, there has been a remarkable success of LLMs in many language-related tasks [DCLT18, RWC⁺19, RSR⁺20]. Latest large language models [NXP⁺22, TGZ⁺23, CLL⁺23,

TLI⁺23] have demonstrated excellent zero-shot capabilities on common-sense inference [CB23]. This success has prompted research into integrating common-sense knowledge into vision-language tasks. In this regard, some methods [SMV⁺19, SZC⁺20, LBPL19, TB19] initialize the language part of vision-language models from pretrained LLM. Another line of work [CLTB21, LLSH23, ZMKG23, LLWL23a] uses LLM as a decoder to enable vision-to-language generation. Works [LRC⁺23, ZMKG23] adapted visually conditioned LLM for visual captioning and created captioning pseudo-labels for large-scale video data. However, these methods require human-annotated datasets to train a captioning model, while our method does not require any label data and aims to transform free available annotation (ASR subtitles) into textual descriptions.

7.3 METHOD

7.3.1 Problem Statement

Given a dataset of N untrimmed long-term instructional videos V_n with corresponding noisy ASR subtitles S_n , our goal is to create “human-written-like” video captions C_n (with $1 \leq n \leq N$). Note that our task does not assume access to any paired training data $((V_n, S_n), C_n)$. The goal is to create the video captions C_n in a *zero-shot* setting given only videos and subtitles (V_n, S_n) . More formally, for each given video V_n , we also have a set of subtitles of spoken text in the video, $S_n = \{s_{n,j}, t_{n,j}^s, t_{n,j}^e\}_{j \leq |S_n|}$ with their start t^s and end timestamps t^e recognized by ASR-systems. For each video V_n , our goal is to generate dense captions and their timestamps $C_n = \{c_{n,i}, \tau_{n,i}^s, \tau_{n,i}^e\}_{i \leq |C_n|}$, where each caption $c_{n,i}$ describes a segment of the video, that starts at $\tau_{n,i}^s$ and ends at $\tau_{n,i}^e$.

The generated captions aim to serve for video-language or video-language-{other modalities (e.g., audio)} tasks, providing language supervision in the form of human-style captions rather than scrambled noisy ASR subtitles. That enables the potential of collecting large-scale datasets with long-term videos and their dense textual descriptions for free, without human supervision.

7.3.2 Video-Language Retrieval Model

Before we describe our method for generating the HowToCaption dataset, we will briefly recap the video-language retrieval models (V-L model), as it is one of the main use cases for this dataset. Moreover, we also use a V-L model to improve the temporal alignment in the dataset.

We base our video-language retrieval model (V-L model) on the pretrained BLIP image-language dual-encoder model [LLXH22b]. We maintain the architecture of the text encoder $f(c) \in \mathbb{R}^d$ but, following CLIP4CLIP [LJZ⁺22], adapt the image encoder $g(I)$ to a video encoder by averaging image embeddings obtained from uniformly sampled frames of the video: $g(V_n) = \sum_{I \in V_n} g(I) \in \mathbb{R}^d$. Dual-encoder models typically learn a cross-modal embedding space [RKH⁺21d, LJZ⁺22] via training with the symmetric InfoNCE loss [OLV18]. The training is based on a similarity metric (often cosine distance) between embeddings $\rho_{n,i,m} = \text{sim}(f(c_{n,i}), g(V_m))$ scaled by a temperature parameter ν , resulting in the following loss function:

$$L = -\frac{1}{2|B|} \sum_{(n,i) \in B} \left(\log \frac{\exp(\rho_{n,i,n}/\nu)}{\sum_{(m,j) \in B} \exp(\rho_{n,i,m}/\nu)} + \log \frac{\exp(\rho_{n,i,n}/\nu)}{\sum_{(m,j) \in B} \exp(\rho_{m,j,n}/\nu)} \right) \quad (7.1)$$

where B is a batch of training sample indices (n, i) .

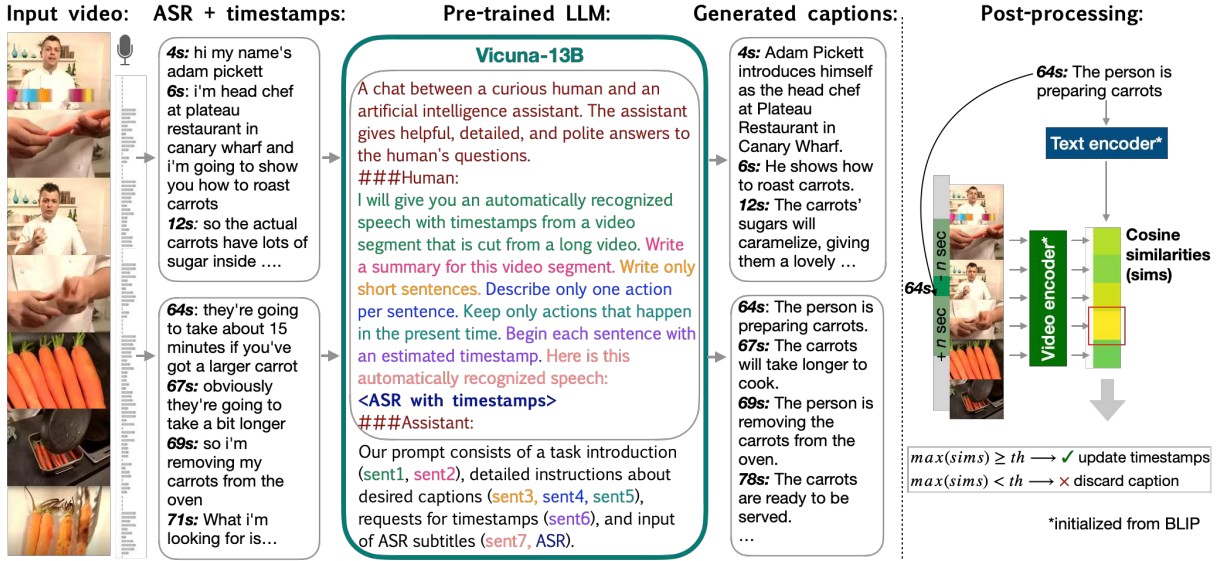


Figure 7.2: **Schematic visualization of the proposed HowToCaption method.** Obtained from the automatic speech recognition system (ASR), subtitles are divided into blocks that contain longer contextual information. A large pretrained language model is then used to generate plausible video captions based on ASR subtitles, along with timestamps for each caption. These generated captions and timestamps are further post-processed to enhance their alignment to the video and filter out captions with low similarity to the corresponding video by leveraging a pretrained text-video model.

7.3.3 HowToCaption Method

To generate captions for the instructional videos, we propose to leverage recent large language models that demonstrate great zero-shot performance in many different tasks formulated with natural language. Namely, we prompt the LLM to read the ASR subtitles of the video and create a plausible video description based on this. Since one subtitle only covers a small part of the video and lacks a global context, we propose to aggregate multiple subtitles together with their timestamp information. Then, we task the LLM to create detailed descriptions based on the entire input and estimate timestamps for each generated sentence.

The overview of our approach is shown in Figure 7.2. For each video, we first slice a given sequence of subtitles into blocks that contain long context information about the video. Then, the ASR subtitles of each block are summarised into a video caption using the LLM that we prompt with our task description. The LLM also predicts timestamps for each sentence in the video caption, which we further refine in our post-processing step based on similarities of a caption sentence to video clips in the neighboring area of predicted timestamps.

LLM Prompting. For our language prompt (shown in Figure 7.2), we leverage the same “main” prompt for the LLM, as in the Vicuna-13B model [CLL⁺23]: “A chat between a curious human and...” that defines the requirement for LLM to give a helpful answer to our questions. Then, we describe our request, what data we need to process, and how it should be processed: “I will give you an automatically recognized speech...”. We found structuring the prompt in the way that the task description given at the beginning of the prompt and the long ASR input S_n at the end is beneficial. Then, we give detailed instructions about how to process ASR subtitles. We

found that instructions such as “Write only short sentences” or “Describe only one action per sentence” are beneficial, as they encourage the creation of concise captions that better match the video content. The instruction “Keep only actions that happen in the present time” is intended to filter out unrelated chats, advice, or comments from the captions; we observed that it also resulted in performance enhancements. Lastly, we request the model to predict a timestamp for each generated caption and, finally, input timestamps + ASR subtitles that need to be processed. The LLM response follows the start timestamp + caption format given in the prompt and, therefore, can be automatically parsed with a simple script into a set of captions and timestamps $C_n = \{c_{n,i}, \tau_{n,i}^s, \tau_{n,i}^e\}_{i \leq |C_n|}$, where we assign $\tau_{n,i}^e = \tau_{n,i}^s + \Delta_{\text{sec}}$, where Δ_{sec} is a constant video clip length parameter (number of seconds). Please see Section 7.4.3 and Appendix E.2.3 for a detailed evaluation of these choices. In Appendix E.1, we also discuss an extension of our HowToCaption method to vision language models that additionally allows grounding captions on visual content. However, we found that this does not improve caption quality.

Post-processing: Alignment & Filtering. ASR subtitles often temporally misalign with video content [HXZ22, MZA⁺19] (e.g., a speaker describes something only after it was shown in a video). Captions derived solely from ASR also face this issue. Therefore, inspired by the TAN method [HXZ22] that automatically predicts the alignability of subtitles and matching timestamps, we further improve our obtained captions with an alignment & filtering post-processing step (Figure 7.2). To this end, we utilize the video-language encoder model (f, g) . Given a generated caption $c_{n,i}$ and its start and end timestamps $(\tau_{n,i}^s, \tau_{n,i}^e)$ that corresponds to a part of the video clip $V_n^{[\tau_{n,i}^s, \tau_{n,i}^e]}$, we use the V-L model to compute alignment similarity scores $\rho_{n,i}(\delta) = \text{sim}\left(f(c_{n,i}), g(V_n^{[\tau_{n,i}^s + \delta, \tau_{n,i}^e + \delta]})\right)$ between the caption and video clips with time offsets $\delta \in \mathbb{Z}, |\delta| \leq T$ around predicted timestamps. Then we *align* the caption to the video clip by finding the best offset around the timestamp $\delta_{n,i}^* = \arg \max_{\delta \in \{-T, \dots, T\}} \rho_{n,i}(\delta)$ and *filter* out pairs if $\rho_{n,i}(\delta_{n,i}^*) < \kappa$, where κ is a similarity score threshold.

To further improve the alignment of captions, we perform multiple rounds of alignment & filtering. In practice, we found that the improvement after two rounds is marginal. For subsequent rounds, we finetune (c.f. Section 7.3.2) the V-L model on the aligned & filtered video-captions pairs $\{(v_i, c_i)\}$, resulting in new alignment scores $\rho'_{n,i}(\delta)$. Since finetuning V-L models often leads to forgetting, we employ two modifications in the finetuning and second alignment processes. First, during finetuning, we add regularization

$$L_{\text{align}} = \alpha \frac{1}{2|B|} \sum_{(n,i) \in B} (\text{sim}(f(c_{n,i}), f^*(c_{n,i})) + \text{sim}(g(V_n), g^*(V_n))) \quad (7.2)$$

where f^* and g^* denote frozen text and video encoders, α is a regularization weight, and $(n, i) \in B$ represents the samples batch B . This regularization prevents the model from forgetting [HPL⁺19]. Then, during alignment & filtering, we use the average of the similarities of the finetuned and original model. We show an impact of these modifications in Appendix E.2.4.

7.3.4 HowToCaption Dataset

We apply the proposed HowToCaption approach to 1.2M long-term instructional videos and ASR subtitles of the HowTo100M dataset and obtain the HowToCaption dataset. By prompting the Vicuna-13B model, we obtain $\sim 70\text{M}$ initial captions. The compute cost for prompting 1.2M videos is $\sim 12\text{k}$ GPU-hours on NVIDIA A40. After alignment & filtering (details in Section 7.4.2)



Figure 7.3: **Examples of video-caption pairs from our HowToCaption dataset.** ASR subtitles with only noisy supervision for the video are converted from spoken- to written-language-style captions. Note that some details in the generated captions are taken from a longer context, see Figure E.2 in the Appendix for a full example.

we obtain 25M high-quality video-caption pairs. We show examples from our HowToCaption dataset in Figure 7.3. We note that generated captions follow different text styles, e.g., the first and the second examples contain a long description of an object and its actions, the third describes the process, and the last one is instruction. The average length of the captions is 9 words. We provide additional examples, statistics, failure case analyses, and user studies in Appendix E.2.1.

7.4 EXPERIMENTAL EVALUATION

To evaluate the proposed HowToCaption dataset for large-scale pretraining of vision-language models, we train a T-V model as described in Section 7.3.2 on the HowToCaption dataset and assess its zero-shot video-text retrieval performance on four widely recognized and diverse video-text benchmarks: YouCook2 [ZXC18], MSR-VTT [XMYR16], MSVD [CD11], and LSMDC [RRS15]. While the YouCook2 dataset consists of instructional cooking videos and might be considered as an in-domain benchmark for the HowToCaption dataset, the other datasets encompass a broader range of topics and video types, including non-instructional YouTube videos and movies. To evaluate the properties of HowToCaption dataset in comparison with other large-scale pretraining datasets, we also train our T-V model on the HowTo100M [MZA⁺19], HowTo100M with step labels [LPB⁺22], HTM-AA [HXZ22], VideoCC3M [NSS⁺22], and WebVid2M [BNVZ21b] datasets and compare zero-shot text-video retrieval performance.

7.4.1 Datasets and Metrics

Pretraining Datasets. **HowTo100M** is a dataset of 1.2M instructional videos with ASR subtitles. We consider three versions of annotations of this dataset: *Sentencified HowTo100M* [HXZ22], with pre-processed ASR subtitles by structuring them into full sentences; *HowTo100M with Distant Supervision* [LPB⁺22], where ASR subtitles were linked to WikiHow [KW18] step descriptions via distant supervision; and *HTM-AA* [HXZ22], an auto-aligned (AA) version of HowTo100M, where subtitle timestamps were adjusted to improve alignment to videos and non-alignable subtitles were discarded. **WebVid2M** [BNVZ21b] is a large open-domain dataset of 2.5M

Prompt	YouCook2		MSR-VTT		MSVD		LSMDC		Average	
	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓
x_1 : Here is an automatically recognized speech from a video: <ASR with timestamps>. x_2 : Write a synopsis for this video. x_3 : Begin each sentence with an estimated timestamp.	37.5	22.5	71.0	3	80.5	2	37.3	30	56.6	14.4
< x_1 > < x_2 > x_4 : Write only short sentences. < x_3 >	39.3	20.5	71.4	3	81.0	2	36.5	32.5	57.1	14.5
< x_1 > < x_2 > < x_4 > x_5 : Describe only one action per sentence. < x_3 >	39.8	20	71.0	3	80.9	2	37.2	30.5	57.2	13.9
< x_1 > < x_2 > < x_4 > < x_5 > x_6 : Keep only actions that happen in the present time. < x_3 >	39.5	19.5	71.6	3	81.2	2	37.9	29	57.6	13.4
< x_1 > x_2' : Write a summary for this video. < x_4 > < x_5 > < x_6 > < x_3 >	40.4	19	71.4	3	81.4	2	37.1	30	57.6	13.5
x_1' : Here is an automatically recognized speech from a video segment that is cut from a long video: <ASR with timestamps> x_2'' : Write a summary for this video segment. < x_4 > < x_5 > < x_6 > < x_3 >	40.0	20	72.0	3	81.1	2	37.8	29	57.7	13.5
<i>I will give you an automatically recognized speech with timestamps from a video segment that is cut from a long video. <x_2''> <x_4> <x_5> <x_6> <x_3> Here is this automatically recognized speech: <ASR with timestamps></i>	40.6	19	72.0	3	81.6	2	37.7	30	58.0	13.5

Table 7.1: **Ablation of LLM prompts.** We step by step construct a prompt for an LLM that concisely and in detail describes the caption generation task. To emphasize our incremental adjustments, we label the sentences as x_n (where n is an index). Each prompt consists of some sentences that were already used in previous prompt versions (e.g., < x_1 >, < x_2 >) and new sentences introduced in the current prompt (e.g., x_4 : Write only ...). With each prompt, we obtain 2M video-text pairs from 100k HowTo100M videos that we later use for T-V model training (lower-resource setup). Downstream zero-shot text-video retrieval performance is reported.

of short videos scrapped from the internet with their alt-text. **VideoCC3M** [NSS⁺22] is a dataset of 10M video-text pairs collected by transferring captions from image-text CC3M dataset [CSDS21] to videos with similar visual content.

Datasets for Downstream Tasks. **YouCook2** [ZXC18] is a dataset of instructional cooking videos, where each video clip is annotated with a recipe step. We used 3.5k test set for evaluation. **MSR-VTT** [XMYR16] contains 10k YouTube videos on various topics and human descriptions. Following prior work [BNVZ21b, NSS⁺22, TWL⁺21], we use the 1k test set for evaluation. **MSVD** [CD11] is a dataset of video snippets with their textual summary. The evaluation set consists of 670 videos with 40 captions corresponding to each video. We follow standard practice [BNVZ21b, LJZ⁺22] and count each caption-video pair towards the metrics. **LSMDC** [RRS15] is a collection of video clips from movies with human-written descriptions. The test set consists of 1k video-caption pairs.

Metrics. To evaluate zero-shot text-video retrieval, we use standard Recall@ K metrics where $K \in 1, 5, 10$ (R1, R5, R10) and Median Rank (MR).

Method	YouCook2				MSR-VTT				MSVD				LSMDC				Average			
	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR
No context	11.1	27.9	38.4	21	37.7	62.4	72.6	3	43.3	71.7	80.2	2	16.5	30.4	38.4	30	27.1	48.1	57.4	14
Long context	12.1	30.0	40.6	19	37.9	61.6	72	3	43.9	72.7	81.6	2	16.8	31.4	37.7	30	27.7	48.9	58.0	13.5

Table 7.2: **Effect of a longer context.** For the “no context” option, we predict captions from individual ASR subtitles. With our “long context” option, we input multiple ASR subtitles with timestamps and the model generated captions based on longer context. This ablation is done in lower-resource setup.

Caption Post-processing	YouCook2		MSR-VTT		MSVD		LSMDC		Average	
	R10 $\uparrow$	MR $\downarrow$	R10 $\uparrow$	MR $\downarrow$	R10 $\uparrow$	MR $\downarrow$	R10 $\uparrow$	MR $\downarrow$	R10 $\uparrow$	MR $\downarrow$
Lower bound: original ASR as supervision	39.3	20	61.7	5	77.1	2	31.5	56	52.4	20.8
No post-processing	40.2	18	65.9	4	79.8	2	34.4	40	55.1	16.0
Filtering (using BLIP)	42.5	16	71.2	3	81.7	2	37.4	30	58.2	12.8
Alignment & filtering (using BLIP)	42.4	17	71.7	3	82.2	2	38.5	29.5	58.7	12.9
Alignment & filtering (with ours)	44.1	15	73.3	3	82.1	2	38.6	29	59.5	12.3

Table 7.3: **Effect of alignment & filtering.** With each post-processing variant, we obtain 25M video-text pairs that we later use for T-V model training. Downstream zero-shot text-video retrieval performance is reported.

7.4.2 Implementation Details

As an LLM, we utilize Vicuna-13B-v0 [CLL⁺23]. In Appendix E.1 we additionally experiment with the MiniGPT-4 model [ZCS⁺23] to generate captions from subtitles grounded on visual content. To create the HowToCaption dataset, we leverage subtitles with timestamps released by [HXZ22] (Sentencified HowTo100M), where officially released subtitles [MZA⁺19] for HowTo100M videos were post-processed by structuring them into full sentences. For our T-V model (described in Section 7.3.2), we use ViT-B/16 visual encoder and BERT_{base} textual encoder that are initialized with BLIP_{CapFilt-L} pretrained weights. We uniformly sample 4 frames from a video clip during training and 12 frames during evaluation. For HowToCaption method, we use $T = 10$ seconds offset for alignment and adaptive threshold κ to leave 25M most similar pairs after filtering. Following [CRD⁺21] that found that 8-sec clips are optimal for training on HowTo100M, we set $\Delta_{\text{sec}} = 8$. More implementation details are in Appendix E.3.

7.4.3 Ablation Study

Prompt Engineering. First, we assess the quality of the obtained dataset with respect to different LLM prompts. Since prompting LLM with subtitles from 1.2M videos is resource-intensive, we perform the prompt engineering ablations in a lower-resource setup, where we use a 100k subset of HowTo100M ($\sim 10\%$ of all videos) to create dense captions with the LLM and use the threshold κ to obtain the 2M most confident video-caption pairs. Here, we train the T-V model for 150k iterations and then evaluate zero-shot on downstream tasks. In Table 7.1, we begin with a basic prompt for the LLM, gradually refining it to generate captions more suitable for vision-language tasks. It is essential to recognize that the impact of various prompts on performance can vary across datasets, as certain prompts may yield captions better aligned

Video-Text Training Data	YouCook2				MSR-VTT				MSVD				LSMDC			
	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR
- (zero-shot, with BLIP initialization)	6.1	16.2	23.6	69	34.3	59.8	70.6	3	38.5	65.0	74.0	2	14.7	29.5	36.5	31
HowTo100M with ASRs [HXZ22]	12.2	29.1	39.3	20	30.8	52.6	61.7	5	39.2	68.3	77.1	2	12.9	24.7	31.5	56
HowTo100M with distant supervision [LPB ⁺ 22]	8.3	21.5	30.3	34	28.6	54.0	66.3	5	38.5	68.6	79.4	2	12.1	24.7	32.4	42.5
HTM-AA (auto-aligned ASRs) [HXZ22]	13.4	32.2	43.5	15	29.8	54.1	64.3	4	38.7	68.6	78.7	2	11.9	23.9	30.5	46
HowToCaption (ours)	13.4	33.1	44.1	15	37.6	62.0	73.3	3	44.5	73.3	82.1	2	17.3	31.7	38.6	29
VideoCC3M [NSS ⁺ 22]	5.3	15.1	21.7	84	33.9	57.9	67.1	4	39.6	66.7	76.8	2	14.8	29.4	35.8	33
WebVid2M [BNVZ21b]	7.3	20.7	29.0	46	38.5	61.7	71.9	3	44.5	73.4	82.1	2	17.8	31.2	39.8	25

Table 7.4: **Zero-shot text-to-video retrieval performance of models trained on different video-text datasets.** For each dataset, we train our T-V model and report downstream performance.

Method	V. Encoder	I-T Data	V-T Data	YouCook2				MSR-VTT				MSVD				LSMDC			
				R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR
Nagrani et al. [NSS ⁺ 22]	ViT-B	-	VideoCC3M	-	-	-	-	18.9	37.5	47.1	-	-	-	-	-	-	-	-	-
Frozen-in-Time [BNVZ21b]	ViT-B/16	CC+COCO	WebVid2M	-	-	-	-	24.7	46.9	57.2	7	-	-	-	-	-	-	-	-
CLIP-straight [PQOBTM21]	ViT-B/32	WIT	-	-	-	-	-	31.2	53.7	64.2	4	37.0	64.1	73.8	2	11.3	22.7	29.2	56.5
CLIP4CLIP [LJZ ⁺ 22]	ViT-B/32	WIT	HowTo100M	-	-	-	-	32.0	57.0	66.9	4	38.5	66.9	76.8	2	15.1	28.5	36.4	28
VideoCoCa [YZW ⁺ 22]	ViT-B/18	JFT-3B	VideoCC3M	16.5	-	-	-	31.2	-	-	-	-	-	-	-	-	-	-	-
BLIP [§] [LLXH22b]	ViT-B/16	6 datasets [‡]	-	6.1	16.2	23.6	69	34.3	59.8	70.6	3	38.5	65.0	74.0	2	14.7	29.5	36.5	30.5
BLIP [§] +HowTo100M	ViT-B/16	6 datasets [‡]	HowTo100M	12.2	29.1	39.3	20	30.8	52.6	61.7	5	39.2	68.3	77.1	2	12.9	24.7	31.5	56
Ours	ViT-B/16	6 datasets [‡]	HowToCaption	13.4	33.1	44.1	15	37.6	62	73.3	3	44.5	73.3	82.1	2	17.3	31.7	38.6	29

Table 7.5: **Comparison in zero-shot text-to-video retrieval with dual-encoder baseline methods.** [§]For BLIP, the performance of dual-encoder architecture is reported. [‡]6 datasets = CC3M [CSDS21]+CC12M [CSDS21]+COCO [LMB⁺14]+VG [KZG⁺17] +SBU [OKB11] +LAION [SVB⁺21a]. “V.” = Vision, “I-T” = Image-Text, “V-T” = Video-Text.

with specific downstream tasks. Notably, incorporating key phrases such as “Write only short sentences” or “Describe only one action per sentence” leads to performance improvements on 3 out of 4 datasets. Additionally, the use of the phrase “Keep only actions that happen in the present time” also results in performance enhancements. Furthermore, structuring the task description at the beginning and presenting the data to be processed at the end (the final modification) also boosts performance. We provide more ablations on prompt engineering in Appendix E.2.3.

We also examine the impact of leveraging a longer context for caption prediction. In Table 7.2, we compare caption generation with “no context”, where captions are predicted from individual ASR subtitles. With our “long context” option, we input multiple ASR subtitles with their timestamps, and the model predicts both captions and timestamps based on longer context. We found that using a longer context is beneficial, resulting in an average improvement of 0.6 p.p. in R10, and particularly advantageous for YouCook2 and MSVD.

Alignment & Filtering. Further, we assess the impact of the proposed alignment & filtering procedure on the quality of captions of the acquired dataset in Section 7.4. We examine the performance of the T-V model when trained on differently post-processed versions of the dataset. Remarkably, we discover that the obtained video-caption pairs, even without any post-processing, significantly outperform the original ASR-based supervision. Subsequently, by employing the alignment and filtering procedure to leave only 25M pairs based on video-caption similarities derived from BLIP pretrained weights, we achieve a notable performance enhancement of 3.6 p.p. in R10. Furthermore, alignment & filtering with our proposed finetuning without forgetting yields an additional 0.8 p.p. boost in R10 performance. More ablations can be found in Appendix E.2.4.

Method	Vision Encoder	YouCook2				MSR-VTT			
		R1↑	R5↑	R10↑	MR↓	R1↑	R5↑	R10↑	MR↓
MIL-NCE* [MAS ⁺ 20]	S3D	15.1	38.0	51.2	10	9.9	24.0	32.4	29.5
TAN* [HXZ22]	S3D	20.1	45.5	59.5	7.0	-	-	-	-
MMT [GSAS20]	Transformer	-	-	-	-	-	14.4	-	66
AVLNet [RBH ⁺ 21]	ResNet-152+ResNeXt101	19.9	36.1	44.3	16	8.3	19.2	27.4	47
MCN [CRD ⁺ 21]	ResNet-152+ResNeXt101	18.1	35.5	45.2	-	10.5	25.2	33.8	-
EAO [SCR ⁺ 22]	S3D	24.6	48.3	60.4	6	9.3	22.9	31.2	35
Ours	S3D	25.5	51.1	63.6	5	13.2	30.3	41.5	17

Table 7.6: **Zero-shot text-to-video+audio retrieval.** *Text-video only models.

Method	V. Encoder	I-T Data	V-T Data	YouCook2				MSR-VTT				MSVD			
				B@4	M	R	C	B@4	M	R	C	B@4	M	R	C
SwinBERT [LLL ⁺ 22]	VidSwin-B	-	-	9.0	15.6	37.3	109	41.9	29.9	62.1	53.8	58.2	41.3	77.5	120.6
CLIP4Caption [TWL ⁺ 21]	ViT-B	-	-	-	-	-	-	46.1	30.7	63.7	57.7	-	-	-	-
GIT-B [WYH ⁺ 22]	ViT-B	4 datasets*	-	5.8	12.2	31.5	80.3	46.6	29.6	63.2	57.8	69.3	44.5	81.4	142.6
MV-GPT [SNAS22]	ViViT-B	-	HowTo100M	-	-	-	-	48.9	38.6	64	60.0	-	-	-	-
LAVENDER	VidSwin-B	5 datasets [†]	WebVid2M+12M	-	-	-	-	-	-	-	60.1	-	-	-	150.7
HiTeA [XYX ⁺ 23]	MViT-B	5 datasets [†]	WebVid2M	-	-	-	-	-	-	-	65.1	-	-	-	146.9
mPlug-2 [XYX ⁺ 23]	ViT-B	5 datasets [†]	WebVid2M	-	-	-	-	52.2	32.1	66.9	72.4	69.3	45.1	81.9	148.2
BLIP [LLXH22b]	ViT-B	6 datasets [‡]	-	7.9	15.0	36.0	104.8	49.2	32.2	66.1	65.5	68.0	45.3	82.8	148.6
BLIP + HowTo100M [§]	ViT-B	6 datasets [‡]	HowTo100M	8.6	15.9	37.1	112.9	49.4	32.0	66.2	65.9	69.3	46.2	83.0	151.4
Ours	ViT-B	6 datasets [‡]	HowToCaption	8.8	15.9	37.3	116.4	49.8	32.2	66.3	65.3	70.4	46.4	83.2	154.2
Vid2Seq [YNS ⁺ 23]	ViT-L	-	YT-Temporal-1B	-	-	-	-	-	30.8	-	64.6	-	45.3	-	146.2
VideoCoCa [YZW ⁺ 22]	ViT-G	JFT-3B	VideoCC3M	-	-	-	-	53.8	-	68.0	73.2	-	-	-	-
GIT-2 [WYH ⁺ 22]	DaViT-4.8B	12.9B pairs	-	9.4	15.6	37.5	131.2	54.8	33.1	68.2	75.9	82.2	52.3	88.7	185.4

Table 7.7: **Video captioning results.** We report BLEU@4 (B@4), METEOR (M), ROUGE-L (R), and CIDEr (C). We gray out methods that use a stronger vision backbone and significantly more pretraining data for fair comparison. *4 datasets = CC3M+COCO+VG+SBU, [†]5 datasets = ...+CC12M, [‡]6 datasets = ...+LAION [SVB⁺21a]. [§]Our finetuning. “V.” = Vision, “I-T” = Image-Text, “V-T” = Video-Text.

7.4.4 Comparison to State-of-the-Art

Comparison with Other Web Datasets. In Table 7.4, we assess the pretraining effectiveness of our proposed HowToCaption dataset compared to other web video-language datasets. Specifically, we evaluate different textual annotations of HowTo100M videos: sentencified ASR subtitles [HXZ22], task steps from distant supervision [LPB⁺22], and auto-aligned ASR subtitles [HXZ22]. Additionally, we conduct evaluations on WebVid2M [BNVZ21b] and VideoCC3M [NSS⁺22] datasets. Our findings indicate that the model pretrained on our HowToCaption dataset significantly outperforms models pretrained on other versions of HowTo100M annotations, with an average improvement of 5.2 p.p. in R10. This improvement is most pronounced for the MSR-VTT, MSVD, and LSMDC datasets, which feature full-sentence captions. Interestingly, for the YouCook2 dataset with captions in the form of step descriptions like “cut tomato”, HTM-AA already exhibits a high baseline performance, but our HowToCaption dataset still provides a performance boost. We also observe that the VideoCC3M dataset does not improve the initial BLIP performance on any datasets except for the MSVD. We attribute it to the fact that the VideoCC3M dataset adopts captions from the CC3M dataset [CSDS21] and transfers them to videos, potentially not introducing significantly new knowledge for the BLIP-initialised model since BLIP was pretrained on multiple datasets, including CC3M. On the other hand, WebVid2M demonstrated performance improvements across all datasets, but

our HowToCaption dataset notably outperforms WebVid2M on YouCook2 and MSR-VTT, only underperforming on LSMDC.

Comparison with State-of-the-art in Zero-shot Text-Video Retrieval. In Table 7.5, we also conduct a comparison with zero-shot *dual-encoder* retrieval baselines. It is important to acknowledge that comparing state-of-the-art methods can be challenging due to variations in backbone capacity, training objectives, and other factors. Therefore, we focus on a comparison with dual-encoder models in the zero-shot settings. Additional comparisons with methods that use *re-ranking* can be found in Appendix E.2.2. Nevertheless, it is worth highlighting that our approach consistently outperforms the baseline methods in zero-shot text-video retrieval across all datasets.

Zero-shot Text-to-Video+Audio Retrieval. It is known that instructional video datasets, e.g., HowTo100M, suffer from a high correlation of audio modality to a textual description, therefore hindering building a text-video+audio retrieval system where the video is extended with audio [SCR⁺22, NSS⁺22]. The usage of ASR narrations as supervisory textual description leads retrieval models to primarily perform speech recognition on the audio, hindering true language-audio connections [SCR⁺22]. Therefore, training text-video+audio systems on such datasets usually requires additional regularization, such as shifting audio timestamps or assigning lower weights to the audio loss [SCR⁺22]. Our HowToCaption dataset resolves this issue by providing richer textual descriptions, allowing us to train a text-video+audio retrieval system without regularization. To evaluate this, we train a multimodal Everything-At-Once (EAO) [SCR⁺22] model that learns to fuse any combinations of text, video, and audio modalities on our proposed HowToCaption dataset without any additional regularization and evaluate zero-shot text-video+audio retrieval performance. Table 7.6 shows the proposed model significantly outperforms all baselines and the directly comparable EAO model.

Video Captioning. We further validate our dataset’s potential for video-language pretraining through video captioning. We finetune the BLIP [LLXH22b] model with all architecture blocks, including the image-grounded text decoder, on the HowToCaption dataset with all three BLIP losses: image-text contrastive loss, image-text matching loss, and language modeling loss. We adapt the image encoder to the video encoder similarly to our retrieval model (Section 7.3.2) and use similar hyperparameters (see Appendix E.3). Subsequently, we finetune the model with language modeling loss on the training set of either YouCook2, MSR-VTT, or MSVD datasets and evaluate video captioning performance. Table 7.7 show that our model finetuned on the HowToCaption dataset significantly outperforms directly comparable BLIP model and other comparable baselines on YouCook2 and MSVD and keeps competitive performance on MSR-VTT.

7.5 CONCLUSION

In this chapter, we propose a novel approach, HowToCaption, that transforms freely available ASR subtitles of instructional videos into high-quality captions, enabling the collection of large-scale, high-quality video-language datasets without manual annotation efforts. With this approach, we curate a new large-scale HowToCaption dataset featuring human-style video captions derived from ASR subtitles of the HowTo100M dataset. We demonstrate that our HowToCaption dataset serves as an excellent source for training video-language representation and video captioning models across four text-video retrieval and three video captioning benchmarks. Furthermore, given the separation of textual descriptions from the audio modality in our dataset, it demonstrates efficacy as a valuable resource for text-to-

video-audio tasks. This work demonstrates the potential of LLMs for creating annotation-free, large-scale video-language datasets. In the next chapter, Chapter 8, we investigate the potential of LLMs to address representation biases in video evaluation benchmarks that may hinder accurate assessment of video understanding capabilities.

UNBIASING THROUGH TEXTUAL DESCRIPTIONS: MITIGATING REPRESENTATION BIAS IN VIDEO BENCHMARKS

Contents

8.1	Introduction	106
8.2	Related Work	108
8.2.1	LLMs for Video Understanding	108
8.3	Discovering Representation Biases	108
8.3.1	Definition: Representation Bias	108
8.3.2	Different Representation Biases	109
8.3.3	Method: Unbiasing through Textual Descriptions (UTD)	110
8.4	Benchmarking Datasets	113
8.4.1	Setup	113
8.4.2	UTD-descriptions	113
8.4.3	Concept Representation Bias	113
8.4.4	Common Sense vs. Dataset Bias	114
8.4.5	Temporal Representation Bias	114
8.4.6	UTD-splits	115
8.5	Benchmarking Video Models	115
8.5.1	Setup	115
8.5.2	Results	116
8.6	Conclusion	117

While the previous chapter focused on improving training data quality to enable more robust training of video models, this chapter focuses on improving evaluation benchmark quality to support a more reliable assessment of video understanding ability. We propose a new “Unbiased through Textual Description (UTD)” video benchmark based on unbiased subsets of existing video classification and retrieval datasets to enable a more robust assessment of video understanding capabilities. Namely, we tackle the problem that current video benchmarks may suffer from different representation biases, e.g., object bias or single-frame bias, where mere recognition of objects or utilization of only a single frame is sufficient for correct prediction. We leverage VLMs and LLMs to analyze and debias benchmarks from such representation biases. Specifically, we generate frame-wise textual descriptions of videos, filter them for specific information (e.g. only objects) and leverage them to examine representation biases across three dimensions: 1) concept bias—determining if a specific concept (e.g., objects) alone suffice for prediction; 2) temporal bias—assessing if temporal information contributes to prediction; and 3) common sense vs. dataset bias—evaluating whether zero-shot reasoning or dataset correlations contribute to prediction. We conduct a systematic analysis of 12 popular video classification and retrieval datasets and create new object-debiased test splits for these datasets. Moreover, we benchmark 30 state-of-the-art video models on original and debiased splits and analyze biases in the models. To facilitate the future development of more robust video understanding benchmarks and models, we release: “UTD-descriptions”, a dataset with our rich structured descriptions for each dataset, and “UTD-splits”, a dataset of object-debiased test splits.

This chapter is based on CVPR 2025 paper [SNS⁺25]. As first author, Nina Shvetsova designed the method, implemented it, conducted all experiments and analyses, and was the main writer of the paper.

8.1 INTRODUCTION

Ever since action recognition moved from simple, staged clip classification to real-life video understanding [MLS09, SZS17, KJG⁺11], there has been an ongoing discussion about how much of a system’s performance can be attributed to the recognition of the action itself compared to related or unrelated visual cues such as objects or scene information in the data [HSSK16, CWR22, WR21], namely *object/scene representation bias* (see Figure 8.1). Different strategies have been proposed to separate action recognition from scene and object influences such as segmenting the human figure in a video and placing it against a gray or black background [HSSK16, CWR22] as well as miming actions in unrelated contexts [WR21]. Some works focused on collecting datasets designed to minimize scene or object representation bias [LLV18, BSYS⁺21, GEKM⁺17b]. However, debiased solutions have so far rarely been adopted for benchmarking. Reasons for that may be that modified videos in case of [HSSK16, CWR22] often introduce artifacts and benchmarks with explicitly low scene bias [LLV18, GEKM⁺17b], might be considered out-of-domain relative to standard training setups. Another line of research questions whether current video benchmarks truly require temporal understanding, as single-frame models have shown comparable performance on several benchmarks [BEG⁺22, LBB23]. While new benchmarks are emerging to assess temporal understanding [BTS23, MAM23, CTZ⁺24], *temporal representation bias* in standard benchmarks remains underexplored.

In this work, we propose a new scalable method, *Unbiasing through Textual Descriptions* (UTD), to systematically analyze and mitigate different types of representation bias, such as object bias, in existing video benchmarks through automatic textual descriptions. Since most datasets lack textual narrations of videos, we leverage Vision-Language Models (VLMs) to generate detailed descriptions of video frames. We then use Large Language Models (LLMs) to extract different concepts from those captions, namely *objects*, *activities*, and *verbs* (an overview of this process is shown in Figure 8.3). Based on generated descriptions, we examine representation biases across three independent dimensions. First, *concept representation bias*—assessing if a specific concept (e.g., objects) alone is sufficient for accurate prediction of the video. Second, *temporal representation bias*—evaluating if temporal information is needed for correct prediction or if single-frame information is sufficient. Lastly, we distinguish between *common sense* and *dataset bias*. Common sense bias refers to the implicit association of certain concepts with specific activities, such as associating an object “a piano” with the label “playing piano.” This differs from dataset bias, where spurious correlations arise within the dataset, e.g., “apply makeup” label frequently occurs with objects like “mirror” and “flowers” in the background. We propose to leverage the zero-shot reasoning capabilities of LLMs to assess common sense bias and train a linear classifier for the dataset bias.

In our study, we found that many samples of popular benchmarks have a significant *object representation bias*, meaning that recognizing only objects on the video is already enough to predict the correct label. Therefore, using such data for benchmarking video backbones may give models with strong object recognition ability an advantage (i.e. a shortcut), *unintentionally* hindering video understanding evaluation with respect to temporal reasoning or action recognition. Therefore, by excluding test samples with the highest object bias, we introduce *UTD-splits*, debiased evaluation splits for considered datasets, which we believe provide a

Guess the activity from the following objects:

<i>Video 1:</i> Person, makeup brush, makeup sponge, makeup palette Answer: _____	<i>Video 2:</i> Person, stage, wall, curtain Answer: _____
--	---

Guess the activity from a single frame:

<i>Video 3:</i>  Answer: _____	<i>Video 4:</i>  Answer: _____
---	--

Figure 8.1: Can you guess activity on a video based solely on the objects or a single frame? (The answer is on the next page in Figure 8.2.)

more robust assessment of video understanding capabilities. Importantly, compared to other approaches that rely on modifying videos [HSSK16, CWR22], this approach neither requires any additional processing of evaluation data, as no new videos are generated nor does it require any finetuning to adjust for any domain bias that could be introduced by debiasing video frames at a pixel level.

We apply the proposed UTD method to analyze and debias six action classification datasets, namely UCF101 [SZS17], SomethingSomethingv2 [GEKM⁺17b], Kinetics-400 [KCS⁺17], -600 [CNB⁺18], and -700 [CNHZ19], and Moments In Time [MAZ⁺19], and six text-to-video retrieval datasets, namely MSRVT [XMYR16], DiDeMo [AHWS⁺17], ActivityNet [CHEGCN15a], LSMDC [RRTS15], YouCook2 [DXDC13], and Spoken Moments In Time [MJL⁺21]. Further, we benchmark 30 state-of-the-art video models of six different model designs, namely VideoMAE [TSWW22], VideoMAE2 [WHZ⁺23], All-in-one [WGY⁺23], Unmasked Teacher [LWL⁺23], VideoMamba [LLW⁺24], and InternVid [WHL⁺23], on the original and UTD-debiased splits. We show that debiasing reduces performance saturation on current benchmarks. Considering model performance drops between original and debiased splits also allows us to analyze models on object bias, e.g., in action classification, larger backbones or those trained on more data appear more robust to object bias.

We summarize the contributions of this work as follows.

- We introduce *Unbiasing through Textual Descriptions (UTD)*, a scalable, automated method to analyze and mitigate representation biases in video benchmarks without the need for human annotators in the loop.
- Using UTD, we analyze representation biases across 12 popular classification and retrieval video benchmarks and release UTD-debiased test splits for each, *UTD-splits*, promoting a more robust evaluation of video understanding.
- We benchmark 30 state-of-the-art video models on both the original and UTD-debiased splits to assess their robustness against object bias under varying conditions.
- Finally, we release *UTD-descriptions*, a dataset of detailed structured descriptions (including objects, activities, and verb categories) for all 12 considered datasets, to support further dataset analysis and the development of less biased benchmarks.

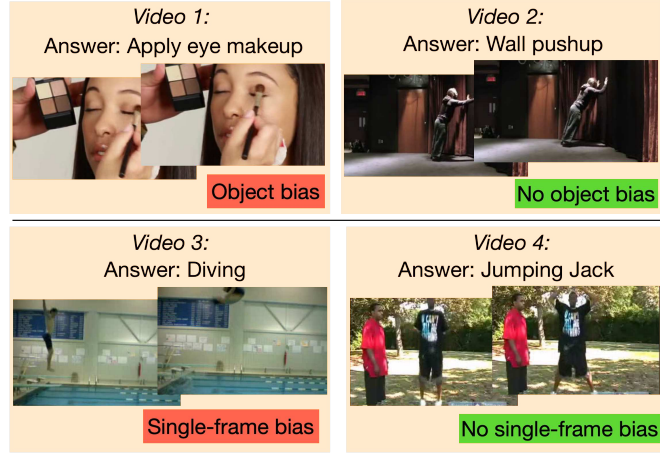


Figure 8.2: The answer to Figure 8.1. Some videos may exhibit an object representation bias (allowing predictions based solely on objects) or single-frame representation bias (solely on a single frame), while others require more information for prediction.

8.2 RELATED WORK

8.2.1 LLMs for Video Understanding

The progress of LLMs prompted a wide range of applications in various fields, including video understanding. Here, one line of work utilizes pretrained LLMs in video-language tasks, such as video captioning [SMV⁺19, ZGY⁺24, WWZ⁺23a], video question answering [ZGY⁺24, LWL⁺23, MRKK24, WWZ⁺23a], text-video retrieval [ZGY⁺24, SKSK23], or dialog-based user interaction [WWZ⁺23a, ZLB23a]. Some works [WHL⁺23, CLW⁺24, SKH⁺24] create new video-language datasets using image/audio captioning models or subtitles and using LLM summarization and reasoning ability to post-process these captions. Other works [ZAI⁺22, LHW⁺23, WLX⁺22, WCL⁺23] use expert perception models to extract information from the video that is later processed by the LLM for tasks such as video captioning. In our work, we leverage VLM- and LLM-generated frame descriptions to evaluate different representation biases in video datasets.

8.3 DISCOVERING REPRESENTATION BIASES

In this section, we define representation bias in Section 8.3.1, outline the considered representation biases in Section 8.3.2, and detail our UTD method in Section 8.3.3.

8.3.1 Definition: Representation Bias

To formalize the definition of representation bias in datasets, we adopt the notation from RESOUND [LLV18].

Let $D = \{v_n\}_{n \leq N}$ be a video dataset D , where each video v_n consists of a sequence of frames $v_n = \{f_{n,i}\}_{i \leq |v_n|}$. A representation ϕ is a function $\phi(v)$ of the video input v . Let $R = \{\phi\}$ be a family of representations that share specific characteristics. For example, a family of single-

frame representations includes representations that encode information from only a single frame of a video $\phi(v_n) = \phi(f_{n,i})$ where $i \leq |v_n|$ or a family of object-based representations encodes only the objects in the video without considering motion or activity information.

The representation ϕ can be used to train different models γ_ϕ for desired tasks, e.g. video classification. Let $M(v_n, \gamma_\phi)$ be the performance (e.g., accuracy) of a model γ_ϕ on a video v_n , then the performance of γ_ϕ on D is

$$M(D, \gamma_\phi) = \frac{1}{N} \sum_{n=1}^N M(v_n, \gamma_\phi). \quad (8.1)$$

We define the performance of a representation ϕ on D as

$$M(D, \phi) = \max_{\gamma_\phi} M(D, \gamma_\phi), \quad (8.2)$$

where the maximum is taken over all models γ_ϕ that are based on the representation ϕ .

When the performance $M(D, \phi)$ of a representation ϕ is high, the dataset exhibits a preference for this representation, indicating that the dataset is more biased toward this representation. In this work, we use $M(D, \phi)$ as a measure of bias towards representation ϕ . However, a high bias toward a specific representation is not necessarily a negative thing. Ideally, a dataset should exhibit the highest bias toward the “ground truth representation”: the representation truly required to solve the task. For instance, in video understanding, one could argue that the optimal representation should capture both static (e.g., objects) and dynamic (e.g., actions) properties, as well as encode temporal aspects. However, if a video dataset used to evaluate video understanding exhibits a strong object or single-frame representation bias, it may unintentionally favor models with strong object recognition or image understanding capabilities, thus hindering the evaluation of true video understanding. Therefore, in this work, we aim to estimate representation biases in popular video benchmarks.

8.3.2 Different Representation Biases

In our work, we consider different types of representation biases with respect to three independent aspects:

Concept Representations: First, we consider representations that encode four different concept categories: *objects* $\phi_o(v_n)$, *activities* $\phi_a(v_n)$ (actions along with tied objects), *verbs* $\phi_v(v_n)$ (actions alone, without objects), and *objects+composition+activities* $\phi_d(v_n)$ (objects, their spatial compositions in the video, and activities), see an example in Figure 8.3. We aim to assess biases in classification and retrieval benchmarks toward particular concepts. For example, in retrieval, we ideally would like to see that benchmarks are not biased solely toward objects or activities.

Temporal Representations: Second, we consider the bias of single-frame representations compared to representations that consider multiple frames. Let ϕ_x be one of the objects ϕ_o , activities ϕ_a , verbs ϕ_v , or objects+composition+activities ϕ_d representations. Similarly to [WXW⁺16], we consider the following options to encode information from multiple frames: *middle-frame* refers to the case that a video is represented only by a single middle frame of this video, e.g., $\phi_{x,\text{mid}}(v_n) = \phi_x(f_{n,|v_n|/2})$, *max-score frame* refers to the case that a video is represented using a single frame $\phi_{x,\text{max}}(v_n) = \phi_x(f_{n,i})$ where the frame $f_{n,i}$ has the highest performance score $M(f_{n,i}, \phi_x)$ over all frames, *average-over-frames* refers to the case that the video is represented as an average representation of frames ignoring the order of frames $\phi_{x,\text{avg}}(v_n) = \frac{1}{|v_n|} \sum_{i \leq |v_n|} \phi_x(f_{n,i})$, and, finally, *sequence-of-frames* refers to the case that the video representation encodes concepts and their order in the frames $\phi_{x,\text{seq}}(v_n) = \phi_x(f_{n,1}, f_{n,2}, \dots, f_{n,|v_n|})$.

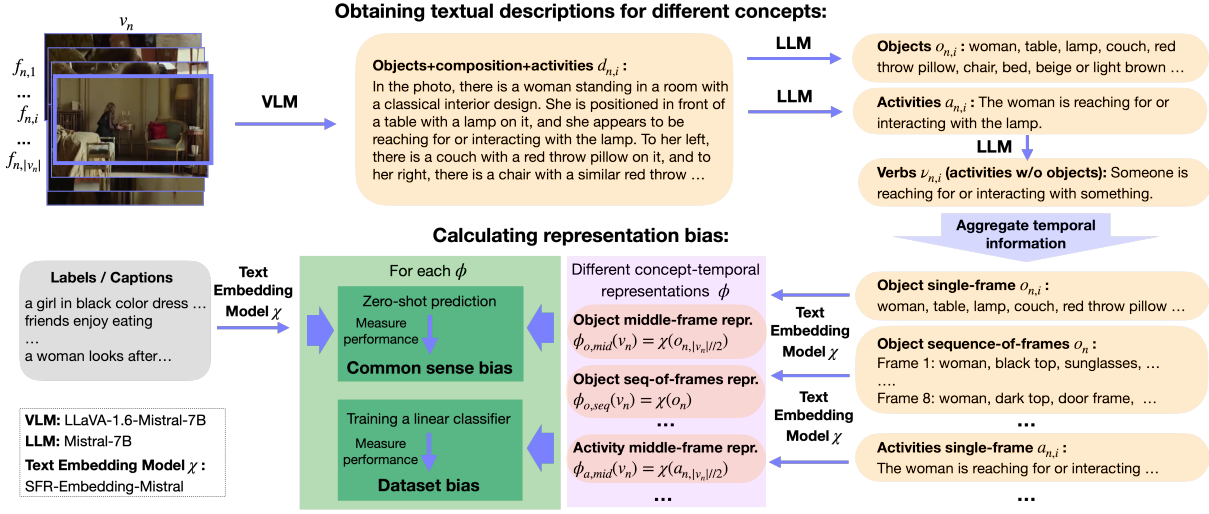


Figure 8.3: **The proposed UTD method** involves generating textual descriptions of different concepts in video frames using VLMs and LLMs, combining them in various temporal configurations, and evaluating the performance of these concept-temporal representations with strong text embedding models. For each representation, we distinguish between common sense bias, which relies on zero-shot reasoning by text embedding models, and dataset bias, assessed using a linear model trained on the dataset’s training set.

Common Sense Bias vs. Dataset Bias: Finally, for any concept-temporal representation ϕ (e.g., object middle-frame representation $\phi_{o,mid}$), we can consider two types of representation biases: *common sense bias* or *dataset bias*. Common sense representation bias refers to bias based on “common sense reasoning”, e.g., a video featuring an object “piano” would most likely be classified under the “playing piano” activity label. Formally, in Equation (8.2), this bias $M(D, \phi)$ is measured by taking the maximum performance only over zero-shot models γ_ϕ that have not seen the correlations in the training set of D . We distinguish this from *dataset bias*, which refers to biases caused by spurious correlations within the dataset, e.g., where certain objects are unintentionally associated with specific labels, such as “mirror” or “flower bouquet” frequently appear with the “apply makeup” label. In this case, in Equation (8.2), we measure dataset bias by taking the maximum performance over models γ_ϕ trained or finetuned on the training set of D .

We note that the concept, temporal, and common sense vs. dataset bias dimensions are independent, allowing us to combine them in any configuration. For example, we can evaluate object middle-frame common sense bias or activity sequence-of-frames dataset bias.

8.3.3 Method: Unbiasing through Textual Descriptions (UTD)

Since measuring bias for every representation within a family (e.g., a family of object representations) is infeasible, we approximate it by considering a single representative from the family [LLV18] and propose a way to precisely control what each representation encodes by using textual descriptions.

The key idea of our method is to use textual descriptions as an intermediate video representation. This approach allows us to easily control which concepts in the video are included or excluded in the representation and to combine information into different temporal repre-

sentations. Namely, to obtain representation ϕ that encodes only a specific concept category, we propose to encode visible concepts of this category into textual descriptions, e.g., encode objects with a list of visible objects, e.g., “*table, lamp, ..*”. This approach, compared to cropping or inpainting [HSSK16, CWR22], allows us to isolate specific concepts more precisely, avoiding information leakage from cropped areas or adding artifacts caused by imperfect inpainting. Additionally, textual descriptions allow us to use off-the-shelf LLMs with strong common sense reasoning abilities to estimate common sense bias. Since metadata with visible objects, activities, etc., is usually not available, we propose using recent powerful VLMs and LLMs to extract this information.

In the next sections, we first describe our method of obtaining textual description with respect to different concept and temporal categories (Section 8.3.3.1), then we describe our method of discovering common sense and dataset biases (Section 8.3.3.2), and lastly, we describe our method of unbiasing benchmarks (Section 8.3.3.3).

8.3.3.1 Obtaining Textual Descriptions

While current VLMs [LLL⁺24, LLLL23, LLWL23b], trained to perform complex image reasoning, are able to generate detailed descriptions of video frames, they still face challenges when prompted to directly extract specific concepts from a frame, e.g., only visible objects. Therefore, we decomposed the task of concept extraction into two parts: extraction of detailed frame description using VLMs and then extraction of different concepts from frame description using LLMs (see Figure 8.3).

Obtaining Textual Descriptions Using VLMs. Given train/test videos $v_n = \{f_{n,i}\}_{i \leq |v_n|}$, $n < N$ of the dataset, we feed each frame $f_{n,i}$ of each video v_n into the VLM model d to generate a detailed textual description $d_{n,i} = d(f_{n,i})$ (Figure 8.3). We found that without complex prompt engineering or few-shot in-context learning strategies [BMR⁺20] VLMs produce long, detailed descriptions. While our method is not restricted to a specific VLM, for reproducibility and feasibility, we select an open-sourced strong middle-scale VLM model, LLaVA-1.6-Mistral-7B [LLL⁺24] with the prompt: “*Describe the objects’ relationships in the photo.*” Since the resulting textual descriptions include objects, their composition, relationships, and visible activities, we use them as *objects+composition+activities* descriptions.

Extracting Concepts Categories Using LLMs. Each of the obtained detailed descriptions $d_{n,i}$ is further fed to the LLM ζ to extract only *objects* $o_{n,i} = \zeta(d_{n,i}, p_o)$, only *activities* $a_{n,i} = \zeta(d_{n,i}, p_a)$ information, with p_o, p_a as respective prompts. Then, to obtain information about *verbs*, we task the LLM with a prompt p_v to delete all object information from activities $v_{n,i} = \zeta(a_{n,i}, p_v)$. Since such text processing requires understanding what objects and activities are, our prompt includes a few examples following a few-shot in-context learning strategy [BMR⁺20]. We provide all prompts in Appendix F.2.

Aggregating Temporal Information. We use the obtained textual descriptions $x_{n,i}$ of video v_n (where $x_{n,i}$ is one of $o_{n,i}, a_{n,i}, v_{n,i}$, or $d_{n,i}$) in different temporal setups: first, we keep them as single frame descriptions that might be further aggregated when transformed into vector representations (e.g. *average-over-frames*), as we describe in Section 8.3.3.2, second, we combine them into *sequence-over-frame* x_n setup with the template: “*Frame 1: $x_{n,1}$ Frame 2: $x_{n,2}$... Frame $|v_n|$: $x_{n,|v_n|}$* ” (Figure 8.3). Since *objects+composition+activities* descriptions $d_{n,i}$ when combined with sequence-over-frame templates become too long, we summarize them to 15-word descriptions $d'_{n,i}$ via an LLM before inputting them into the sequence prompt.

8.3.3.2 Calculating Representation Biases

After obtaining a specific concept-temporal representation ϕ , we assess common sense and dataset bias for dataset D toward ϕ by measuring the performance $M(D, \phi)$, as stated Equation (8.2), where maximum is taken over zero-shot models γ_ϕ for *common sense* bias and over finetuned models γ_ϕ for *dataset bias* (see Section 8.3.2). Since maximizing over all models is infeasible in practice, we design strong models for common sense and dataset-based reasoning by utilizing state-of-the-art LLMs. Our ablation study verifies our design choice (see Appendix F.1.3).

Common Sense Bias. We leverage recent strong LLM-based text embedding models [WYH⁺23, MLJ⁺24]. Such models are trained on large datasets to encode text into vector representations that can be used for various tasks, such as information retrieval. We encode our textual descriptions and all class labels/captions using the pretrained text embedding model χ . Specifically, we prompt the model with the relevant concept and task. For example, we use prompts like “Given a list of objects visible on the video frame, retrieve the activity depicted in this video” for object description and “Given an activity, retrieve a video frame description that may depict this activity” for class labels in action classification. We use cosine similarity between obtained embeddings for zero-shot classification/retrieval. Prompts are detailed in Appendix F.2.

Dataset Bias. For action classification datasets, we evaluate dataset bias, as action classification models are often tested in finetuned setups and might exploit dataset representation biases learned from training data. For this we use a linear logistic regression model trained on our strong text embeddings used in the common sense bias analysis and train model on the official train splits of the datasets.

Metrics. For the performance measure $M(D, \phi)$, we use accuracy for action classification and recall@1 and text-to-video retrieval as the most popular metrics for these tasks.

8.3.3.3 Unbiasing Datasets

In Section 8.4, we find that many video datasets exhibit a significant object representation bias. To support robust video model evaluation, we construct *UTD-splits*, object-debiased test splits, by excluding object-biased samples from the original test sets. Since our method allows us to measure the performance of the representation $M(v_n, \phi)$ on individual samples v_n , we exclude samples that are classified or retrieved correctly ($M(v_n, \phi) = 1$) based on *object sequence-of-frame* representation. For text-to-video retrieval, we debias with respect to *common-sense bias*, but for action classification, we debias with respect to *dataset bias* since video models are mostly tested in finetuned setup. To minimize the impact of random fluctuations in the process, we use three different prompts for labels/captions in the text embedding models and train three linear models for each classification dataset using bootstrapped training sets (we provide details and analysis in Appendix F.2.3) and exclude only samples when all nine models agreed on correct classification/retrieval. Note that the percentage of removed samples is determined automatically based on the extent of object representation bias.

Finally, since debiasing may disproportionately remove certain label classes in classification datasets, we additionally construct balanced UTD splits. While maintaining the total number of removed samples as in the original debiasing method, we adjust the number of samples removed from each class based on their average confidence (across nine models) to preserve the original class proportions.

8.4 BENCHMARKING DATASETS

In this section, we analyze action classification and text-to-video retrieval datasets on different representation biases using our UTD method.

8.4.1 Setup

Datasets. We consider six common **action classification** datasets: UCF101 (UCF) [SZS17], SomethingSomethingv2 (SSv2) [GEKM⁺17b], Kinetics400 (K400) [KCS⁺17], -600 (K600) [CNB⁺18], -700 (K700) [CNHZ19], Moments In Time (MiT) [MAZ⁺19] and six **text-to-video retrieval** datasets: MSRVT (MSR) [XMYR16], YouCook2 (YC2) [DXDC13], DiDeMo (DDM) [AHWS⁺17], LSMDC [RRTS15], ActivityNet (ANet) [CHEGCN15a], and Spoken Moments in Time (S-MiT) [MJL⁺21].

Implementation Details. For all train/test videos, we use $|v_n| = 8$ uniformly sampled frames. We use the LLaVA-1.6-7B-mistral model [LLL⁺24] as VLM, Mistral-7B-Instruct-v0.2 [JSM⁺23] as LLM, and SFR-Embedding-Mistral [MLJ⁺24] as text embedding model. The temperature is set to 0.

8.4.2 UTD-descriptions

Following our UTD pipeline, first, we generate textual descriptions for 12 considered datasets, resulting in *UTD-descriptions* dataset (see qualitative examples Appendix F.3). We conducted a user study to validate the quality of object, activity, and verb predictions. Participants were asked to verify which objects from a provided list were visible in a given image. The study used 100 random frames from 100 videos of the UCF dataset (which has noisy, low-resolution frames that might be an additional challenge to the VLM). Seven participants validated 761 annotated objects in total, where 87.6% (667 out of 761) were identified as visible, with only 94 not selected as visible. To better understand the VLM’s errors, we manually classified these 94 objects into five categories: 1) attribute error (e.g., the object is “right hand” instead of “left hand”), 2) misclassification (the object is present but incorrectly identified as a different object, e.g., a “snowboard” instead of “snow slide”), 3) hallucination, 4) human annotation mistake – the object is visible, and 5) other. We found that only 45 objects were classified as hallucinations, 13 as misclassifications, 20 as “object visible” 8 as attribute errors, and 8 as other (see examples in Appendix F.3). This results in a hallucination rate of only 6%, which we find acceptable for our analysis.

8.4.3 Concept Representation Bias

Further, based on generated descriptions, we evaluate representation bias with respect to four concept categories, namely objects, activities, verbs, and objects+composition+activities, as described in Section 8.3.2. For this, we consider *common sense bias* in *sequence-of-frames* temporal setup in Table 8.1 (all concept-temporal combination results can be found in Appendix F.1.2). It shows that using only object information results in only a slight drop in performance compared to objects+composition+activities, indicating that most datasets can, to a certain extent, be addressed with object information, thus showing an object representation bias. Interestingly, classification based on only objects results in 63.3pp accuracy for UCF101 and 45.9pp for Kinetics400. This may be due to the fact that some action classes are tied to the objects (e.g.

	UCF	SSv2	K400	K600	K700	MiT	MSR	DDM	ActN	LSMDC	YC2	S-MiT
obj+comp+act	66.3	6.4	48.0	44.1	39.0	22.6	36.6	27.2	26.6	17.0	8.4	45.9
objects	63.3 _{-3.0}	5.3 _{-1.1}	45.9 _{-2.1}	41.8 _{-2.3}	37.0 _{-2.0}	21.0 _{-1.6}	32.1 _{-4.5}	27.0 _{-0.2}	24.8 _{-1.8}	13.6 _{-3.4}	7.9 _{-0.5}	29.8 _{-16.1}
activities	67.4 _{+1.1}	6.4	45.2 _{-2.8}	41.4 _{-2.7}	36.7 _{-2.3}	21.0 _{-1.6}	25.1 _{-11.5}	21.1 _{-6.1}	21.4 _{-5.2}	14.7 _{-2.3}	8.1 _{-0.3}	41.1 _{-4.8}
verbs	50.8 _{-15.5}	5.8 _{-0.6}	24.8 _{-23.2}	21.4 _{-22.7}	17.6 _{-21.4}	16.2 _{-6.4}	10.5 _{-26.1}	7.0 _{-20.2}	7.4 _{-19.2}	5.5 _{-11.5}	1.2 _{-7.2}	13.1 _{-32.8}

Table 8.1: **Evaluation of concept representation bias.** We report the performance of our model in *sequence-of-frames* temporal setup and *common sense bias* setup and color code with respect to the difference to *objects+composition+activities* (*obj+comp+act*) concepts. It shows that the overall classification performance drops only slightly when classifying actions based on objects, compared to classification and retrieval on verbs only, with SSv2 as the only dataset with less drop when classifying on verbs only.

	UCF			SSv2			K400			K600			K700			MiT		
	obj.	act.	verbs	obj.	act.	verbs	obj.	act.	verbs	obj.	act.	verbs	obj.	act.	verbs	obj.	act.	verbs
common sense bias	63.3	67.4	50.8	5.3	6.4	5.8	45.9	45.2	24.8	41.8	41.4	21.4	37.0	36.7	17.6	21.0	21.0	16.2
dataset bias	80.3	83.4	73.5	13.6	16.0	12.2	60.7	60.7	42.2	57.7	58.4	39.4	49.4	49.7	31.6	25.2	25.4	21.2
	+17.0	+16.0	+22.7	+8.3	+9.6	+6.4	+14.8	+15.5	+17.4	+15.9	+17.0	+18.0	+12.4	+13.0	+14.0	+4.2	+4.4	+5.0

Table 8.2: **Comparison of common sense bias vs dataset bias.** We report the performance of our model in *sequence of frame* temporal setup and color code with respect to difference to *common sense bias*. It shows that the overall performance growth is significant for all concepts, including object concepts, indicating that datasets might contain a significant correlation between labels and visible objects.

“playing piano”), making visible objects equally or more reliable indicators of a class than activities. This becomes clear when we consider the verbs-based representation. Here, the performance drops significantly for most action classification datasets, except for SSv2 and MiT. Also, in the case of text-to-video retrieval, it shows that objects are better indicators than activities for some datasets (MSRVTT, DiDeMo, ActivityNet), and activities are better indicators for others (YouCook2, LSMDC, S-MiT).

8.4.4 Common Sense vs. Dataset Bias

We further consider common sense bias compared to dataset bias for the action classification datasets in Table 8.2. Overall, across all benchmarks and concepts, dataset bias is significantly higher than common sense bias, and the results indicate that considered benchmarks also contain a dataset-specific object bias in the training set. The performance on UCF using only objects increases from 63.3pp to 80.3pp, and even on SSv2, which was developed to minimize object representation bias, using only correlation of seen objects in the train set to the classes, we were able to predict 13.6pp of videos correctly. This significant increase based on available object information only shows that some actions might be connected to unrelated objects that hint at a certain class without being directly associated with it.

8.4.5 Temporal Representation Bias

Finally, we consider representation bias with respect to the temporal aspect in Table 8.3. Namely, we consider which temporal representations, single frame vs. multiple frames, best-score frame vs. middle frame, as well as average-over-frames vs. sequence-of-frames

	UCF	SSv2	K400	K600	K700	MiT	MSR	DDM	ActN	LSMDC	YC2	S-MiT
middle-frame	61.3	6.0	39.7	35.6	31.1	20.1	23.4	19.8	13.5	12.7	6.1	35.5
max-score frame	<u>66.5</u>	<u>7.4</u>	<u>48.0</u>	<u>43.3</u>	<u>38.7</u>	21.4	33.4	29.1	<u>21.5</u>	18.3	8.9	46.1
avg.-over-frames	66.7	<u>6.8</u>	<u>46.6</u>	42.1	37.1	23.4	31.9	26.5	26.6	16.4	<u>8.6</u>	43.8
seq.-of-frames	66.3	6.4	48.0	44.1	39.0	<u>22.6</u>	36.6	<u>27.2</u>	26.6	<u>17.0</u>	8.4	<u>45.9</u>

Table 8.3: **Evaluation of temporal representation bias.** We report the performance of our model in *objects+composition+activities* concepts setup and *common sense bias* setup.

	UCF	SSv2	K400	K600	K700	MiT	MSR	DDM	ActN	LSMDC	YC2	S-MiT
Fleiss’ kappa	86.1	77.8	91.2	90.7	92.5	91.2	92.0	91.7	90.9	92.5	91.3	93.2
% object-biased samples	73	7	55	55	48	24	28	23	20	11	6	36
% of full test split in UTD-split	27	93	45	45	52	76	72	77	80	89	94	64

Table 8.4: **UTD-splits statistics.** Fleiss’ kappa coefficient measures the agreement among 9 models used to create UTD-splits.

embeddings, perform best. For all datasets except ActivityNet, max-score-frame performance is comparable ($< 3.5\%$ difference) to multiple frames performance (average of frames/sequence over frames). Moreover, for SSv2, Kinetics400, DiDeMo, LSMDC, YouCook2, and S-MiT, we found that max-score-frame is a better indicator of class/captions than all frames.

8.4.6 UTD-splits

Since our analysis reveals significant object bias in benchmarks, we generate object-debiased UTD-splits for considered benchmarks, as described in Section 8.3.3.3. Namely, we remove object-biased samples where nine models that use different prompts in the text embedding model or trained on the bootstrapped train set uniformly agree that the sample can be predicted correctly based solely on objects. Table 8.4 summarizes the percentage of object-biased samples identified in the datasets, the level of model agreement (measured by Fleiss’ kappa coefficient, which is notably high), and the number of samples retained from the original datasets.

8.5 BENCHMARKING VIDEO MODELS

In this section, we benchmark state-of-the-art video encoders on considered datasets and their UTD-splits.

8.5.1 Setup

Video Models. We consider state-of-the-art video encoders with varying model sizes and pretraining configurations with open-sourced weights. For **action classification**, we benchmark 24 different variants of VideoMAE [TSWW22], VideoMAE2 [WHZ⁺23], All-in-one [WGY⁺23], Unmasked Teacher (UMT) [LWL⁺23], VideoMamba [LLW⁺24], and InternVid [WHL⁺23] families. For **text-to-video retrieval**, we consider 17 video-language models of UMT [LWL⁺23], VideoMamba [LLW⁺24], and InternVid [WHL⁺23] families. In total, we analyze 30 models

	UCF	UCF- UTD- split	UCF- UTD- s. balanced	SSv2	SSv2- UTD- split	SSv2- UTD- s. balanced	K400	K400- UTD- split	K400- UTD- s. balanced
videomae-B-K400	89.2	81.3 _{-7.9}	82.8 _{-6.4}	54.2	52.1 _{-2.1}	53.0 _{-1.2}	64.3	45.7 _{-18.6}	49.5 _{-14.8}
videomae-B-UH	88.4	79.2 _{-9.2}	80.4 _{-8.0}	56.2	54.0 _{-2.2}	54.9 _{-1.3}	64.3	45.4 _{-18.9}	49.6 _{-14.7}
videomae-L-UH	95.7	90.9 _{-4.8}	91.0 _{-4.7}	67.3	65.4 _{-1.9}	66.2 _{-1.1}	75.8	59.3 _{-16.5}	63.3 _{-12.5}
videomae-H-UH	95.2	90.9 _{-4.3}	91.0 _{-4.2}	67.4	65.5 _{-1.9}	66.3 _{-1.1}	76.5	59.5 _{-17.0}	63.7 _{-12.8}
videomae-v2-B-K710-fnK710	99.0	98.1 _{-0.9}	97.6 _{-1.4}	57.1	54.7 _{-2.4}	55.9 _{-1.2}	85.0	71.9 _{-13.1}	75.5 _{-9.5}
allinone-B-WV2M+CC	84.5	63.1 _{-21.4}	72.9 _{-11.6}	26.2	22.5 _{-3.7}	24.7 _{-1.5}	66.9	42.6 _{-24.3}	50.8 _{-16.1}
allinone-B-WV2M+HT	81.4	57.7 _{-23.7}	68.3 _{-13.1}	22.7	19.3 _{-3.4}	21.3 _{-1.4}	61.4	36.9 _{-24.5}	45.2 _{-16.2}
allinone-B-WV2M+HT+CC+YTT+	81.3	58.5 _{-22.8}	66.2 _{-15.1}	22.0	18.8 _{-3.2}	20.7 _{-1.3}	61.2	36.4 _{-24.8}	44.6 _{-16.6}
umt-B-K710	91.3	76.8 _{-14.5}	82.4 _{-8.9}	50.6	47.9 _{-2.7}	49.1 _{-1.5}	77.7	58.3 _{-19.4}	63.7 _{-14.0}
umt-B-K710-fnK710	99.0	96.8 _{-2.2}	97.1 _{-1.9}	49.4	46.6 _{-2.8}	48.1 _{-1.3}	85.6	72.4 _{-13.2}	76.2 _{-9.4}
umt-L-K710	95.7	87.2 _{-8.5}	89.2 _{-6.5}	59.4	57.0 _{-2.4}	58.1 _{-1.3}	84.0	68.4 _{-15.6}	73.0 _{-11.0}
umt-L-K710-fnK710	98.9	96.7 _{-2.2}	97.1 _{-1.8}	57.9	55.5 _{-2.4}	56.7 _{-1.2}	89.0	78.7 _{-10.3}	81.6 _{-7.4}
videomamba-vm-K400	90.9	74.6 _{-16.3}	81.0 _{-9.9}	35.9	32.6 _{-3.3}	34.4 _{-1.5}	75.3	54.4 _{-20.9}	60.5 _{-14.8}
videomamba-vm-5M	93.7	82.0 _{-11.7}	86.7 _{-7.0}	48.5	45.6 _{-2.9}	46.9 _{-1.6}	77.5	58.1 _{-19.4}	64.0 _{-13.5}
videomamba-vm-17M	93.2	80.5 _{-12.7}	84.3 _{-8.9}	47.4	44.7 _{-2.7}	45.9 _{-1.5}	77.7	58.1 _{-19.6}	63.6 _{-14.1}
videomamba-vm-25M	94.3	83.0 _{-11.3}	86.5 _{-7.8}	48.7	45.9 _{-2.8}	47.1 _{-1.6}	78.4	59.1 _{-19.3}	64.6 _{-13.8}
internvid-B-10M-FLT	94.0	81.1 _{-12.9}	84.7 _{-9.3}	48.1	45.2 _{-2.9}	46.7 _{-1.4}	78.6	59.3 _{-19.3}	64.9 _{-13.7}
internvid-B-200M	94.5	82.9 _{-11.6}	85.2 _{-9.3}	54.4	51.8 _{-2.6}	52.8 _{-1.6}	80.0	61.7 _{-18.3}	66.8 _{-13.2}
internvid-L-10M	93.2	80.2 _{-13.0}	85.3 _{-7.9}	52.6	50.0 _{-2.6}	51.1 _{-1.5}	77.4	57.4 _{-20.0}	63.8 _{-13.6}
internvid-L-WebVid10M	94.1	82.5 _{-11.6}	86.3 _{-7.8}	54.9	52.4 _{-2.5}	53.4 _{-1.5}	78.9	59.7 _{-19.2}	65.7 _{-13.2}
internvid-L-10M-DIV	94.8	83.2 _{-11.6}	87.6 _{-7.2}	51.7	48.8 _{-2.9}	50.1 _{-1.6}	80.3	62.1 _{-18.2}	67.4 _{-12.9}
internvid-L-10M-FLT	95.5	86.3 _{-9.2}	88.4 _{-7.1}	53.6	50.9 _{-2.7}	52.1 _{-1.5}	81.9	64.7 _{-17.2}	69.6 _{-12.3}
internvid-L-50M	95.2	85.5 _{-9.7}	88.4 _{-6.8}	60.6	58.2 _{-2.4}	59.3 _{-1.3}	81.3	63.7 _{-17.6}	69.1 _{-12.2}
internvid-L-200M	96.8	90.5 _{-6.3}	91.9 _{-4.9}	63.1	61.0 _{-2.1}	61.9 _{-1.2}	84.6	69.4 _{-15.2}	73.9 _{-10.7}

Table 8.5: **Benchmarking video models in action classification.** We report accuracy on full test/val sets and our debiased UTD-splits. The accuracy differences with respect to the full test/val sets are color-coded.

and list all considered variants in Appendix F.2.4.

Implementation Details. To evaluate action classification performance in a standard aligned setup, we follow the VideoGLUE [YGZ⁺23] benchmark of using a frozen video backbone and lightweight pooling head as a test of the strength of video model representation. To adapt each video backbone for a specific dataset, we create a model that 1) uses frozen video backbone to extract spatio-temporal features, and then 2) uses a single attention pooling layer [YGZ⁺23] to aggregate spatio-temporal features into a single vector representation and 3) a single linear projection layer to classify on the required number of classes. We train the model on a full train split of the corresponding dataset. For video model evaluation in text-to-video retrieval, we reported their zero-shot performance on our debiased splits. We follow the corresponding model recipes in using re-ranking and how many examples are reranked. We report the accuracy for classification and the recall@1 for text-to-video retrieval. Additional details are provided in Appendix F.2.4.

8.5.2 Results

We evaluate the performance of video models for action classification on the UCF, SSv2, and K400 datasets and their UTD-splits in Table 8.5 and for text-to-video retrieval on the MSRVT, DiDeMo, and ActivityNet datasets and their UTD-splits in Table 8.6. Performance for other datasets is reported in Appendix F.2.4. Across all models, performance drops notably on debiased splits (except for SSv2), suggesting that current models strongly rely on object

	MSR	MSR- UTD- split	DDM	DDM- UTD- split	ANet	ANet- UTD- split
umt-b-5M	30.0	20.5 _{-9.5}	30.2	20.6 _{-9.6}	28.6	22.3 _{-6.3}
umt-b-17M	35.6	26.3 _{-9.3}	37.7	27.6 _{-10.1}	34.2	27.3 _{-6.9}
umt-b-25M	35.3	24.8 _{-10.5}	34.2	24.6 _{-9.6}	25.1	19.7 _{-5.4}
umt-l-5M	34.8	24.9 _{-9.9}	33.5	21.8 _{-11.7}	34.8	28.7 _{-6.1}
umt-l-17M	43.6	31.1 _{-12.5}	46.3	35.6 _{-10.7}	45.9	38.7 _{-7.2}
umt-l-25M	42.3	30.6 _{-11.7}	43.6	33.5 _{-10.1}	36.7	30.4 _{-6.3}
videomamba-vm-5M	33.3	23.0 _{-10.3}	37.1	27.1 _{-10.0}	37.1	30.1 _{-7.0}
videomamba-vm-17M	34.9	25.5 _{-9.4}	40.6	28.9 _{-11.7}	40.4	33.0 _{-7.4}
videomamba-vm-25M	34.9	25.5 _{-9.4}	41.4	30.5 _{-10.9}	41.1	33.8 _{-7.3}
internvid-B-10M-FLT	37.9	25.4 _{-12.5}	28.6	17.2 _{-11.4}	24.4	18.8 _{-5.6}
internvid-B-200M	38.1	24.7 _{-13.4}	30.2	19.4 _{-10.8}	26.2	20.1 _{-6.1}
internvid-L-10M	26.7	18.7 _{-8.0}	22.6	15.6 _{-7.0}	21.5	16.3 _{-5.2}
internvid-L-WV10M	26.5	17.6 _{-8.9}	22.2	13.8 _{-8.4}	23.1	17.5 _{-5.6}
internvid-L-10M-DIV	37.3	24.2 _{-13.1}	26.9	15.9 _{-11.0}	23.2	17.7 _{-5.5}
internvid-L-10M-FLT	38.7	26.3 _{-12.4}	29.2	18.8 _{-10.4}	24.5	19.0 _{-5.5}
internvid-L-50M	32.4	22.0 _{-10.4}	26.5	18.3 _{-8.2}	24.4	18.0 _{-6.4}
internvid-L-200M	38.2	24.8 _{-13.4}	30.3	20.2 _{-10.1}	28.7	22.1 _{-6.6}

Table 8.6: **Benchmarking video-language models in text-to-video retrieval.** We report accuracy on full test/val sets and our debiased UTD-splits. The accuracy differences with respect to the full test/val sets are color-coded.

representation bias. Evaluation on UTD debiased splits shows a higher range of scores of models, revealing that some models (e.g. all-in-one models) were more heavily relying on object bias rather than utilizing action/motion cues for video understanding. Further, we can observe that the best performance flips for some models for the debiased case compared to the full test set, e.g., for videomae-B-K400 compared to videomamba-vm-K400 on the UCF dataset.

Additionally, by comparing the performance decline between the original datasets and the UTD-splits, we can assess models’ sensitivity to object bias. Overall, we see in the case of *action classification*, Table 8.5, that VideoMAE models are most robust to object bias, whereas Allinone models are most impacted by object bias. We also note that performance drops correlate between the UTD-split and UTD-split-balanced versions. We can further observe that larger backbones/architectures, and to a certain extent, more training data, lead to a reduced bias compared to smaller models, e.g. InternVid and VideoMAE. This pattern does not hold for *text-to-video retrieval*, shown in Table 8.6. Here, larger backbones and backbones trained with more data turn out to have higher object bias than smaller models or models trained with less data (see InternVid). We attribute this behavior, to the fact that retrieval models leverage information from both the vision as well as the text backbone. Thus, if larger text backbones also learn more object properties or a larger vocabulary of objects, this might further encourage object-centering learning in the visual stream as well.

8.6 CONCLUSION

In this chapter, we presented a new way to examine representation biases in video classification and retrieval datasets by leveraging recent, powerful VLMs and LLMs with no human involvement. Our investigation confirms that most video datasets are heavily focused on object

recognition. Based on our bias analysis, we propose a method to reduce object bias in these benchmarks, enabling more robust video understanding evaluation. With that, we are able to benchmark state-of-the-art video models and show an analysis of the object bias in these models. We hope that our findings, together with the presented benchmark, offer valuable insights for the future development of more robust video understanding benchmarks and models.

To summarize, in this part of the thesis, Part III, we address the data aspect of model development by improving both training and evaluation datasets without human annotation. In the next part, Part IV, we continue developing models without human supervision and focus on dense prediction capabilities, namely spatial and spatio-temporal localization.

IV

GROUNDING WITH MULTIMODAL MODELS WITHOUT HUMAN SUPERVISION

In the final part of the thesis, we study how to enable dense properties in representations trained without any human annotations using only free multimodal supervision. Specifically, we address the tasks of spatial and spatio-temporal grounding in videos.

In Chapter 9, we propose a new framework for spatio-temporal action grounding that learns solely from noisy multimodal supervision, specifically instructional videos accompanied by subtitles. To assess this task under challenging, realistic conditions, we also release a new benchmark with dense spatio-temporal annotations in long, untrimmed, multi-action instructional videos.

In Chapter 10, we further explore how to uncover the activity-grounding abilities of frozen pretrained multimodal models without additional training or finetuning and propose the first training-free method for spatial action grounding.

WHAT, WHEN, AND WHERE? SELF-SUPERVISED SPATIO-TEMPORAL GROUNDING IN UNTRIMMED MULTI-ACTION VIDEOS FROM NARRATED INSTRUCTIONS

Contents

9.1	Introduction	122
9.2	Related Work	124
9.2.1	Supervised Spatio-temporal Grounding	124
9.2.2	Multimodal Self-supervised Learning	125
9.3	Method	125
9.3.1	General Setup	125
9.3.2	Representation Guided Frame Sampling	126
9.3.3	Local Representations for Spatial Localization	127
9.3.4	Global Representations For Temporal Localization	128
9.3.5	Inference for Spatio-Temporal Grounding	128
9.4	GroundingYoutube Benchmark	128
9.5	Experimental Evaluation	129
9.5.1	Datasets	129
9.5.2	Baseline Methods	130
9.5.3	Downstream Tasks	130
9.5.4	Comparison to State-of-the-Art	131
9.5.5	Ablation Study	132
9.5.6	Qualitative Results	134
9.6	Conclusion	134

Spatio-temporal grounding describes the task of localizing events in space and time, e.g., in video data, based on verbal descriptions only. Models for this task are usually trained with human-annotated sentences and bounding box supervision. This work addresses this task from a multimodal supervision perspective, proposing a framework for spatio-temporal action grounding trained on loose video and subtitle supervision only, without human annotation. To this end, we combine local representation learning, which focuses on leveraging fine-grained spatial information, with a global representation encoding that captures higher-level representations and incorporates both in a joint approach. To evaluate this challenging task in a real-life setting, a new benchmark dataset is proposed, providing dense spatio-temporal grounding annotations in long, untrimmed, multi-action instructional videos for over 5K events. We evaluate the proposed approach and other methods on the proposed and standard downstream tasks, showing that our method improves over current baselines in various settings, including spatial, temporal, and untrimmed multi-action spatio-temporal grounding.

This chapter is based on the CVPR 2024 paper [CSR⁺24]. As a second author, Nina Shvetsova contributed to weekly scientific discussions, writing of the paper, and creating the figures.

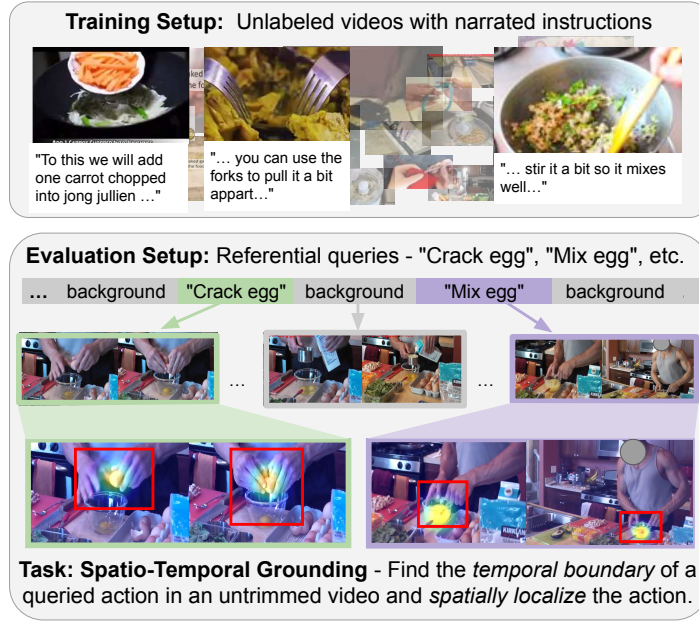


Figure 9.1: **Learning Spatio-temporal grounding in untrimmed videos.** In training, we learn from unlabeled videos without human annotation. In evaluation, we perform spatio-temporal grounding using an action description such as “crack egg” as a query. The model needs to localize both the action’s temporal boundary and spatial region in the long untrimmed video. We visualize the heat-map from the annotation points as well as derived bounding boxes.

9.1 INTRODUCTION

Spatio-temporal grounding (STG) describes the challenging task of locating events in space and time within video data based on text referential expressions. Methods in this field usually rely on a combination of spatio-temporal bounding box annotation, together with a human-generated caption, describing the visual content of the bounding box [YMS⁺22, JyLYM22], which limits their generalizability beyond the given training scenario. Compared to that, as a second line of work, multimodal self-supervised learning tries to leverage “free” data sources, such as video and automatic speech recognition (ASR) captions from large-scale instructional videos to learn representations without human annotation [MZA⁺19, MAS⁺20, ARS⁺20, AYQ⁺21, CRD⁺21]. The resulting models achieve state-of-the-art performance on zero-shot tasks such as cross-modal video retrieval or classification and also for zero-shot temporal action segmentation and detection based on free text queries [ZAC⁺19a, KIRG19, TDR⁺19, CRD⁺21, SCR⁺22], but usually lack spatial localization abilities. A third line of work focuses on label-free spatial grounding, e.g. by training on image-caption [AKB⁺19, YGW⁺22, LZZ⁺22, WLL⁺22, ZYZ⁺22] or video-caption pairs [TPS⁺21, SXGX19]. The goal is to correctly localize a referential expression in an image or each video frame, e.g., via a bounding box or a heatmap. However, those methods are not optimized to detect whether an event is present in a video. The assumption is thus that the evaluated expression is visible in the image or in all video frames.

The following work aims to bring together those ideas to address the task of spatio-temporal action grounding from multimodal supervision in untrimmed videos. We propose a grounding approach that uses video-text pairs based on ASR transcripts in instructional videos and learns

the spatial representation of free-text events as well as their temporal extent, as shown in Figure 9.1. To this end, we leverage two different representations of the visual data: a global feature representation based on full-frame information to define the temporal extent of an event and a local representation based on frame-wise grid features for spatial localization. The motivation for this dualism is that while the local representation captures the spatial correlations between vision and text input, this can be too fine-grained to learn a holistic representation of the frame, while the global representation can be assumed to capture a more compact, aggregated view compared to local data and thus to provide a more reliable cue for the task of temporal localization. However, compared to the hand-annotated video-caption setup of most spatio-temporal grounding methods, the ASR text can be noisy as not all comments refer to visible events. Further, as there is only a loose temporal correlation, the described activities might not be precisely aligned, can be scattered over multiple frames, or not be present at all [MAS⁺20, HXZ22]. Therefore, we propose to specifically select frames to capture only those useful for training. To this end, we look for frames that match the vocabulary of the respective text, leveraging a selection strategy by Sinkhorn optimal transport [Cut13]. This allows us to train a model that can localize actions in space and time within videos without labeling supervision.

To evaluate spatio-temporal grounding in untrimmed videos, a new benchmark, GroundingYouTube, is proposed. It is based on the existing MiningYouTube dataset [KIRG19] and extended with spatio-temporal localization information. This setup differs from other benchmarks such as [TPS⁺21, CMLW19, ZLC18b] in two ways: first, by using multiple center point annotations, it focuses on the grounding of referential actions itself instead of interacting humans or objects which are usually labeled; second, the dense annotations of multiple actions in the video allow us to benchmark action grounding in long, realistic untrimmed videos compared to existing, often pre-clipped benchmarks[CMLW19, ZZZ⁺20]. The benchmark provides queries for 512 different event types and over 5K spatio-temporal annotations, as shown in Figure 9.1. A comparison of current datasets is shown in Table 9.1.

To evaluate the proposed approach as well as the new benchmark, the system is trained on the HowTo100M dataset [MZA⁺19] and compared to state-of-the-art methods based on full, weak, and self-supervision for spatial and temporal, as well as combined spatio-temporal grounding tasks. It shows that existing methods usually do well in one of the two aspects, spatial or temporal grounding. In contrast, the proposed method can combine spatial and temporal aspects.

We summarize the contributions of this work as follows:

- We propose a framework for spatio-temporal grounding in untrimmed videos based on weakly aligned multimodal supervision without human annotation, employing a combination of global and local representation learning to learn the spatio-temporal extent of actions.
- To facilitate this task, we propose a frame selection strategy based on Sinkhorn-Knopp Optimal transport that improves the quality of the acquired learning samples, leading to more effective supervision.
- We provide a new benchmark and annotations to evaluate this challenging problem on real-world multi-action instructional video data.

Dataset	Annotation	
	Spatial	Temporal
V-HiCo	object bb + human bb	-
AVA-Kinetics	object bb + human bb	-
ActivityNet entities	object bb + human bb	-
THUMOS14	-	action boundaries
ActivityNet	-	action boundaries
HACSSegment	-	action boundaries
YouCook II	-	multi-action boundaries
Cross-Task	-	multi-action boundaries
COIN	-	multi-action boundaries
EPIC KITCHENS-100	-	multi-action boundaries
Ego4D	-	multi-action boundaries
JHMDB51-21	human tubes	-
UCF101-24	human tubes	action boundaries
Daly	human tubes	action boundaries
Vid-STG	human tubes	action boundaries
HC-STVG	human tubes	action boundaries
AVA	human tubes	action boundaries
YouCook-Interactions	action bb	-
GroundingYoutube (ours)	action bb + center points	multi-action boundaries

Table 9.1: **Comparison of spatial, temporal, and spatio-temporal grounding datasets.** V-HiCo [LDT⁺21], AVA-Kinetics [LTR⁺20], THUMOS14 [IZJ⁺17], ActivityNet [CHEGCN15b], ActivityNet entity [ZKC⁺19], HACSSegment [ZTTY19], YouCook II [ZXC18], Cross-Task [ZAC⁺19a], COIN [TDR⁺19], EPIC KITCHENS-100 [DDF⁺22], Ego4D [GWB⁺22], JHMDB51-21 [JGZ⁺13], UCF101-24 [SZS12] Daly [WMS16], Vid-STG [ZZZ⁺20], HC-STVG [TLL⁺21], AVA [GSR⁺18], YouCook-Interactions [TPS⁺21].

9.2 RELATED WORK

9.2.1 Supervised Spatio-temporal Grounding

Spatio-temporal Grounding refers to the problem of localizing a sequence of bounding boxes (a spatio-temporal tube) for a target object described by an input text. This problem has been addressed by various approaches TubeDETR [YMS⁺22], STCAT [JyLYM22], STVGBert [SYX21], STVGBert [SYX21], STGVT [TLL⁺21], STGRN [ZZZ⁺20], Visil [KZPPK19]. These methods rely on proposal networks such as Faster R-CNN [RHGS15] or MDETR [KSL⁺21] to predict bounding box coordinates for learning text-to-region interaction. All those approaches rely on supervised training with the human-annotated sentence and bounding box supervision, provided, e.g., by datasets such as VidSTG [ZZZ⁺20], HC-STVG [TLL⁺21]. While those datasets provide a temporal aspect, temporal detection is usually limited to identifying the start and end frame of a single action in a video. Compared to that, an untrimmed setting usually comprises multiple actions in a longer video that can be separated by longer background sequences. This conceptually differs from previous works [CMLW19] that typically use short videos of around 5-10 seconds. Other datasets such as ActivityNet entities [ZKC⁺19] provide only bounding boxes for noun phrases in the captions, namely the objects, which is related to object detection task and does not capture any spatial or temporal extent of actions.

9.2.2 Multimodal Self-supervised Learning

The field of multimodal self-supervised learning aims to learn data representations by leveraging large amounts of unlabeled data with multiple modalities. Early works [WBU11, FCS⁺13] started by projecting images and text into a joint visual-language embedding space, where embeddings of semantically similar pairs are close. Those ideas have now grown into systems such as MIL-NCE [MAS⁺20] using the HowTo100M dataset [MZA⁺19] to train a video-language embedding space from 1.2 million instructional videos paired with text descriptions from ASR. Follow-up works, including [ARS⁺20, AYQ⁺21, RBH⁺21, CRD⁺21, SCR⁺22] show that using videos without annotation enables an effective multimodal embedding space via contrastive learning.

Based on those advantages, approaches started to address the problem of **Spatial Video Grounding** from multimodal self-supervised aiming to identify spatial locations in a *trimmed* video based on text descriptions without the need for bounding box annotation during training. One of the early works studied this task in the context of weakly supervised learning where we learn grounding with human-annotated captions of the video [ZLC18b]. In this context, works [TPS⁺21, SXGX19] have focused on object grounding benchmarks such as YouCook2-BoundingBox [ZXC18], which provides bounding box annotations for visible objects in cooking videos. Other works such as GLIP [LZZ⁺22], RegionCLIP [ZYZ⁺22], and others [YHW⁺22, ZZH⁺22] combine the principles of large-scale vision language training with bounding box finetuning on object detection datasets [GDG19, LMB⁺14]. Recently, the YouCook-Interactions dataset [TPS⁺21] and CoMMA [TPS⁺21] have been proposed for the spatial grounding of objects and actions with multimodal self-supervision from HowTo100M videos. These works assume that the video is temporally clipped with respect to the grounding phrase.

Compared to that, **Temporal Video Grounding** aims to determine the set of consecutive frames corresponding to a text query in an *untrimmed* video [JLZ⁺14, SPA⁺21, CTY21], thus predicting temporal boundaries of action instances. Recent work such as MIL-NCE [MAS⁺20], MCN [CRD⁺21], and VideoCLIP [LHH22] utilize large-scale pretraining for grounding actions temporally via text-to-frame similarity on video datasets such as MiningYouTube [KIRG19] or CrossTask [ZAC⁺19a] without proposals. However, these methods lack spatial localization ability [ZXH⁺20, ZZZL21].

9.3 METHOD

9.3.1 General Setup

The goal of the proposed method is to temporally and spatially localize actions based on free-text queries in untrimmed videos. To this end, two representations are learned, a local and a global one. We start with narrated video clips, each associated with a corresponding visual representation and text narration. Namely, for each clip $\mathcal{X} = \{\mathcal{V}, \mathcal{S}\}$, let $\mathcal{V}$ stand for the video clip and $\mathcal{S}$ for the text narration sentence generated by the automatic speech recognition (ASR) system. Each clip $\mathcal{V}$ consists of $U \times N$ spatio-temporal tokens $\{v_{u,n}\}$, where $u \in \{1, \dots, U\}$ represents the number of frames in the video and $n \in \{1, \dots, N\}$ represents the number of spatial grid region tokens or features in a frame. The text sentence $\mathcal{S}$ consists of K words $\{s_1, \dots, s_K\}$. We represent localized features by the tokens from each modality, and the global features $\{V, S\}$ are acquired either by mean-pooling over the local features (S3D) or by using

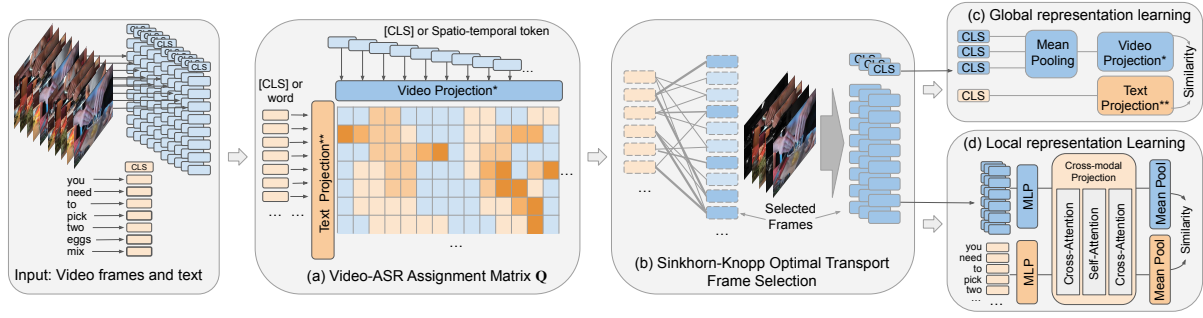


Figure 9.2: **Spatio-temporal grounding approach.** (a) We aim to select frames with groundable objects and actions. To this end, projected text features are matched with respective frame features. (b) Sinkhorn optimal transport is then leveraged to optimize the selected frames wrt. the text input. (c) Based on the selected frames, a global representation is learned to allow for temporal localization, as well as (d) a local representation to ground the action description in the spatial region.

the [CLS] token from the transformer (CLIP) as in [RKH⁺21b]. We learn transformations $f : V \rightarrow \mathbb{R}^d$ to a d -dimensional representation $f(V) \in \mathbb{R}^d$ from the global representation V , and $g : S \rightarrow \mathbb{R}^d$, to produce similar d -dimensional text global embeddings: $g(S) \in \mathbb{R}^d$. Similar to $\{f, g\}$, we note $\{f', g'\}$ to be the transform for localized features, where local features $\{v, s\}$ are also projected as d -dimensional representations.

9.3.2 Representation Guided Frame Sampling

Learning from multimodal self-supervision is challenging since the narration is likely to be noisy, thus containing more information than the actual task descriptions due to poor temporal alignment or cut scenes [HXZ22], which is the key differences between weakly supervised vision-caption grounding and multimodal self-supervised grounding. This work pursues a frame selection strategy to improve object grounding and temporal alignment during training. We start from a longer sequence U , where $U > T$, which includes the video frames before and after the ASR boundaries that could contain actions or objects in the sentence. Our goal is to find T frames out of the U frames that are most relevant to the actions and objects in the sentence S . We formalize it as an optimal transport problem utilizing the Sinkhorn-Knopp algorithm [Cut13].

Optimal transport for text-to-frame assignment. To acquire the optimal assignment from word features to video frames, an assignment matrix $\mathbf{Q}$ is computed from each video and ASR pair as shown in Figure 9.2(a). This cross-model optimal transport mechanism is applied to assignment $\mathbf{Q}$ from the projected cross-model similarity $\mathbf{P}$ between word tokens and each video frame, where $\mathbf{P} = g(S) \otimes f(V)^\top \in \mathbb{R}^{K \times U}$. To compute the assignment matrix, the text and video projection layers from the global representation in Figure 9.2(c) are used to project multimodal features into a common space for feature similarity calculation. We investigate various granularities of the features where we compute the similarity between the text features at the word (local) / sentence (global) level and the visual feature at frames (global) / spatiotemporal tokens (local) level to acquire $\mathbf{P}$, as shown in Table 9.5. To ensure that the word-to-frame assignment contains more diversity instead of just saturated assignments to a single video frame, we add a constraint by Sinkhorn that requires label assignments to be equally distributed across various video frames representing diverse object/action concepts.

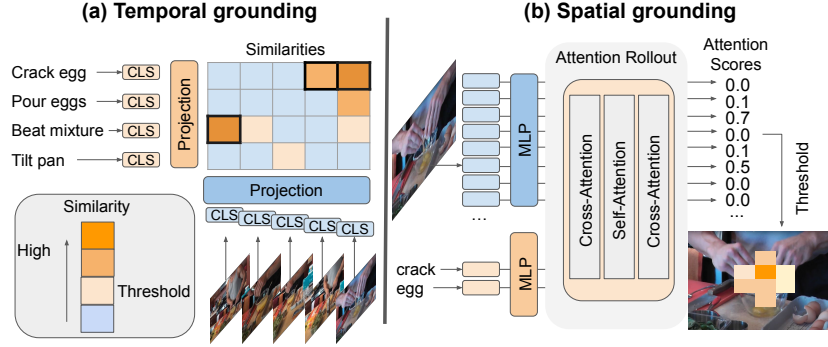


Figure 9.3: **Spatio-temporal inference.** Both representations are used for spatio-temporal grounding: Starting by predicting the action boundary, spatial grounding is performed on the selected frames using the predicted label to find corresponding regions.

Details of the Sinkhorn optimal transport are included in Appendix G.1.

9.3.3 Local Representations for Spatial Localization

To capture multimodal interaction with finer granularity, we need to learn the projection between tokenized features as shown in Figure 9.2(d). We extract spatio-temporal region features v_{tn} from the video. Also, we extract word features s_k which represents the feature from word k . All tokenized features are projected through a linear layer. To compute attention between the tokenized features, we stacked two cross-modal attention layers with a self-attention layer in the middle, as illustrated in Figure 9.2 (d). Cross-modal attention is computed similar to the standard attention mechanism [LCH⁺18]. Given a spatio-temporal token v_{tn} from a video, we compute the attention score to all of the words s_k , where $k \in \{1, \dots, K\}$ in the ASR sentence $\mathcal{S}$ by $\alpha_{tnk} = \frac{\exp(e_{tnk})}{\sum_{k=1}^K \exp(e_{tnk})}$ in the same video clip, where $e_{tnk} = \text{cosine}(v_{tn}, s_k)$. We then acquire a contextual video token feature $\bar{v}_{tn} = \sum_{k=1}^K \alpha_{tnk} s_k$, which encoded text contextual information. Note that the contextual vector is represented by aggregating the representations from the other modality. Follow the standard self-attention computation [VSP⁺17] K, Q, V represent the features for the keys, queries, and values as: $\text{Attn}(K, Q, V) = \text{softmax}\left(\frac{(Q^\top K)}{\sqrt{d_k}}\right) V$ where d_k is the dimension of the key. In our case, we feed each contextual feature $\{\bar{v}_{tn}, \bar{s}_k\}$ right after the first cross-attention layer to the K, Q, V to acquire its self-attended representation. The localized attention model was trained using contrastive loss. To represent the video clip $\mathcal{V}$ and ASR sentence $\mathcal{S}$, we mean-pool over the spatio-temporal tokens in video $\bar{V} = \frac{1}{TN} \sum_{r=1}^{TN} \bar{v}_r$, and words $\bar{S} = \frac{1}{K} \sum_{k=1}^K \bar{s}_k$ respectively. Let $(\bar{V}^{(l)}, \bar{S}^{(l)})$ be the l -th training example pair. We adopt the Noise Contrastive Estimation (NCE) loss [GH10] and the localized attention losses $\mathcal{L}_{Local}$:

$$-\frac{1}{B} \sum_{l=1}^B \left[\left(\log \frac{e^{\bar{V}_l \cdot \bar{S}_l - \delta}}{e^{\bar{V}_l \cdot \bar{S}_l - \delta} + \sum_{\substack{k=1 \\ k \neq l}}^B e^{\bar{V}_k^{imp} \cdot \bar{S}_l}} \right) + \left(\log \frac{e^{\bar{V}_l \cdot \bar{S}_l - \delta}}{e^{\bar{V}_l \cdot \bar{S}_l - \delta} + \sum_{\substack{k=1 \\ k \neq l}}^B e^{\bar{V}_l \cdot \bar{S}_k^{imp}}} \right) \right] \quad (9.1)$$

where B is the batch. $\bar{V}_k^{imp}$ and $\bar{S}_k^{imp}$ represent imposter samples, and δ is a margin hyperparameter.

9.3.4 Global Representations For Temporal Localization

We learn the global representation of a video clip and a sentence by contrastive loss, as shown in Figure 9.2(c). We again use the NCE loss function [GH10]. The global contrastive loss $\mathcal{L}_{Global}$ follows the formulation as Equation 9.1 while using the global representations V and S , which are the [CLS] tokens or mean-pooled features from both modalities, instead of the local representations. Projecting the global features to the same space ensures that the features across different modalities are comparable. Since global representations encode information from the entire video, it is essential in encoding temporal information for the later downstream tasks. The final model is optimized by the sum of both losses as $\mathcal{L}_{Final} = \mathcal{L}_{Local} + \mathcal{L}_{Global}$.

9.3.5 Inference for Spatio-Temporal Grounding

To perform spatio-temporal grounding on untrimmed videos, we start from temporal action detection as shown in Figure 9.3. Given a pool of possible action descriptions on the left and an untrimmed video, we perform feature similarity matching using the global representation ([CLS] token or mean-pooled feature) per frame with a threshold τ to filter backgrounds. We pick the action class with the largest similarity score per frame. Later, we use the predicted action class and feed it into the local representation branch to compute spatial grounding. We follow [AZ20] to compute feature similarity between visual tokens and text tokens through the cross-attention and self-attention. In the end, we acquire an attention heatmap for later downstream evaluation. More information on inferencing are in Appendix G.2.3.

9.4 GROUNDING YOUTUBE BENCHMARK

Current downstream datasets either provide spatial [TPS⁺21], or temporal annotation [KIRG19], or spatio-temporal annotation [ZZZ⁺20] but only for short video clips with few frames before and after the action takes place. These datasets do not provide the opportunity to evaluate both aspects, spatial and temporal grounding, in an untrimmed long video manner. We, therefore, extend one of the current benchmarks, MiningYouTube [KIRG19], with 250 untrimmed videos with a duration of around 3-5 minutes and an average of 41.6 labeled action instances per video. Note that each video contains various classes. As the dataset already provides dense temporal annotations, we annotate the respective temporal action segments in the dataset with spatial information.

Annotating the spatio-temporal extent of actions can be challenging as there is no clear visible outline as, e.g., in object annotation, nor is there a unique signal to indicate the temporal begin and end points. Similarly, grounding systems do not produce pixel-exact bounding boxes, but rather indicate regions of interest. Detector-free spatial grounding models [ADA⁺21] address this fuzziness by relying on pointing game accuracy, thus only using the center point of the heat map for evaluation. Lending on this idea, annotators were asked to mark the presumed center point of the action. Compared to bounding boxes, center point annotation can be advantageous because annotators are not visually distracted by object outlines, so it is more likely that the most important region will be selected. We capture five annotations per frame, resulting in a density-based heatmap.

Starting from 5,091 clips showing one of the 512 action classes, we adopt the methodology used for temporal action localization developed in [GSR⁺18] and label one frame per second, resulting in 26,987 labeled frames. We annotated all frames with five repeats per image,

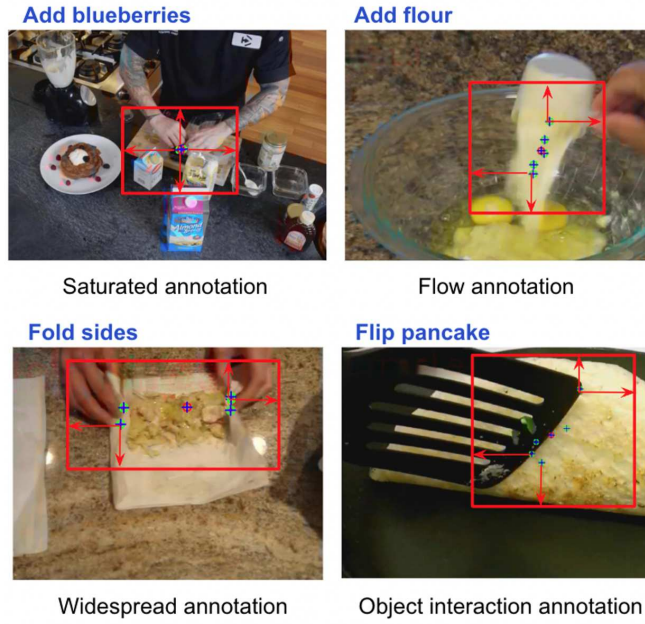


Figure 9.4: **Visualization of the point annotation and automatic bounding box generation from points.** The red point represents the mean of the five annotation points. The points annotation captures diverse patterns in various action types.

resulting in five annotations per frame and 134,935 point labels in total. To evaluate using bounding boxes [KWFS17], we get the union of all annotated points with additional distance to construct the bounding box as shown in Figure 9.4. More information on the annotation process, bounding box derivation, and dataset analysis is provided in Appendix G.4.

9.5 EXPERIMENTAL EVALUATION

9.5.1 Datasets

Training Data: **HowTo100M dataset** contains 1.2M instructional videos along with their corresponding automatically generated speech (ASR). The narrations may be inaccurate and do not always accurately depict the video scene.

Downstream Datasets: **GroundingYoutube (GYT)** is used to evaluate the task of multi-action spatio-temporal grounding as described in Section 9.4. **MiningYoutube (MYT)** [KIRG19] provides temporal annotation and is limited to the domain of cooking instruction videos. **YouCook-Interaction (YC-Inter)** [TPS⁺21] is an extension of the YouCook2 dataset [ZXC18] for cooking instruction providing bounding boxes for 6K selected frames. The bounding boxes usually comprise the hand and the tool mentioned in the respective sentence-wise annotation. To further benchmark on general video domains on the **V-HICO** dataset [LDT⁺21] with 6.5k videos with human-object interaction bounding boxes annotations, and **Daly** action dataset [WMS16], featuring videos consisting of daily actions such as “brushing teeth”.

Method	Backbone	Dataset	Supervision	Mod.	IoU+Point	GroundingYoutube					
						mAP					
						0.1	0.2	0.3	0.4	0.5	0.1:0.5
CoMMA [†] [TPS ⁺ 21]	S3D	HT250K	Self	VT	1.02	2.18	1.72	1.11	0.93	0.37	1.26
MIL-NCE [MAS ⁺ 20]	S3D*	HT100M	Self	VT	4.67	33.94	25.16	12.65	3.42	0.41	15.11
Ours	S3D	HT100M	Self	VT	9.12	42.70	35.49	25.16	16.22	10.05	25.92
GLIP [LZZ ⁺ 22]	Swin-L*	Cap24M	Weak	IT	1.24	2.83	2.10	1.52	0.96	0.37	1.56
CoMMA [‡] [TPS ⁺ 21]	CLIP	HT100M	Self	VT	1.68	3.51	2.32	1.88	0.99	0.40	1.82
CLIP [RKH ⁺ 21b]	CLIP	HT100M	Self	IT	3.59	29.54	22.15	9.16	2.48	0.39	12.74
RegionCLIP [ZYZ ⁺ 22]	ResNet-101*	CC3M	Weak	IT	5.65	35.65	27.43	15.69	4.31	0.86	16.78
Ours	CLIP	HT100M	Self	VT	10.09	42.81	36.05	25.84	17.10	11.35	26.63
Ours	CLIP*	HT100M	Self	VT	11.53	43.64	36.94	26.78	19.45	14.61	28.26
MIL-NCE(temp.)+RegionCLIP(spa.)	-	-	-	VT	9.21	40.54	34.97	22.38	13.79	9.18	22.33

Table 9.2: **Spatio-temporal grounding on GroundingYouTube full videos.** The proposed model learns global representations encoding global information and spatial correspondences across modalities, achieving a better performance in spatio-temporal evaluation compared to models trained on only spatial or temporal grounding. Mod.=Modality (V: video, I: image, T: text), * indicates finetuned backbone, [†] denotes weights shared by the authors, [‡] trained with CLIP backbone.

9.5.2 Baseline Methods

Temporal: MIL-NCE [MAS⁺20] utilizes S3D [XSH⁺18] and word2vec [MCCD13]. CLIP [RKH⁺21b], an image-text model with transformer. **Spatial:** CoMMA [TPS⁺21], SSL model ([†] for weights shared by the author¹ [‡] trained with CLIP); GLIP [LZZ⁺22], RegionCLIP [ZYZ⁺22], SOTA weakly supervised grounding model. **Spatio-temporal:** We construct MIL-NCE+RegionCLIP following the inference pipeline in Figure 9.3. TubeDETR [YMS⁺22] and STCAT [JylyM22] are supervised. More descriptions of the baselines are given in Appendix G.2.1. Details of the implementation and experimental settings can be found in Appendix G.2.2. Inference setups for each baseline are described in Appendix G.2.3.

9.5.3 Downstream Tasks

We considered the following downstream tasks to evaluate spatio-temporal grounding abilities of various models (detailed description is included in Appendix G.2.4):

(i) **Spatio-temporal grounding in untrimmed video** is evaluated on the proposed Grounding Youtube dataset. The entire video and the respective pool of action instructions were provided. The model needs to localize each action step in time (start-time/end-time) and space (location in the video) as described in Figure 9.3. We evaluate in two metrics: **IoU+Pointing game** combines the evaluation setting from the spatial grounding [AKB⁺19] and temporal grounding [KIRG19] metrics. For each video frame, the prediction is correct when the model predicts the correct action for the frame. Also, given the predicted action as a query, the maximum point of the heatmap aims to lie within the desired bounding box. We then compute the Intersection over Union (IoU) over all the predictions with the GT to acquire the final score. We also compute **video mAP** following previous evaluation [GSR⁺18], where we set IoU threshold between GT and predicted spatio-temporal tubes. A prediction is correct when it surpasses the IoU threshold. We then compute the mAP over all classes. We form a 3D prediction mask

¹We thank the authors for providing code and weights.

Method	Backbone	Data	Super.	Mod.	YC-Inter	GroundingYT		V-HICO		Daly	
					Acc	Acc	mAP	Acc	mAP	Acc	mAP
MIL-NCE [MAS ⁺ 20]	S3D*	HT100M	Self	VT	23.67	27.45	8.21	12.65	11.23	13.84	24.23
CoMMA [†] [TPS ⁺ 21]	S3D	HT250K	Self	VT	48.63	47.68	23.38	40.97	21.45	54.48	33.39
Ours	S3D	HT100M	Self	VT	53.98	60.62	44.93	44.32	24.31	66.35	45.93
CLIP [RKH ⁺ 21b]	CLIP	HT100M	Self	IT	14.10	12.50	3.49	29.23	12.51	18.02	27.28
CoMMA [†] [TPS ⁺ 21]	CLIP	HT100M	Self	VT	52.65	47.56	36.42	55.20	34.54	61.06	44.37
RegionCLIP [YZY ⁺ 22]	RN50x4*	CC3M	Weak	IT	51.56	52.84	23.42	57.92	37.82	67.12	48.62
GLIP [LZZ ⁺ 22]	Swin-L*	Cap24M	Weak	IT	52.84	53.62	24.73	66.05	41.17	-	-
Ours	CLIP	HT100M	Self	VT	57.10	55.49	43.12	60.71	39.28	70.08	50.56
Ours	CLIP*	HT100M	Self	VT	58.35	56.98	45.32	62.34	41.56	71.35	52.78
TubeDETR [YMS ⁺ 22]	MDETR	Vid-STG	Full	VT	51.63	53.24	41.76	63.23	40.87	84.21	62.98
STCAT [JylYM22]	ResNet-101	Vid-STG	Full	VT	54.47	55.90	44.21	65.34	41.10	85.42	63.94

Table 9.3: **Video spatial grounding.** We evaluate the accuracy of the pointing game and the mean average precision. We listed CNN-based methods on top and transformer-based methods in the middle. Models learning global representations (MIL-NCE, CLIP) don’t perform well on localization tasks, while our model outperforms other grounding methods. Super.=Supervision, Mod.=Modality (V: video, I: image, T: text), * indicates finetuned backbone, [†] denotes weights shared by the authors, [‡] trained with CLIP backbone.

following Figure 9.3 and compute IoU between our 3D heatmap and 3D tube.

(ii) **Spatial grounding** is given a text description to localize the region in the trimmed video. It is evaluated using the **pointing game accuracy**. If the predicted point lies in the ground truth bounding box, the result counts as a “hit” and counts as “miss” otherwise. The final accuracy is calculated as a ratio between hits to the total number of predictions $\frac{\# \text{ hits}}{\# \text{ hits} + \# \text{ misses}}$. We also report the mean average precision (**mAP**) following the settings from V-HICO [LDT⁺21].

(iii) **Temporal grounding** provides videos with the respective actions and their ordering, including the background. The goal is to find the correct frame-wise segmentation of the video. We follow the inference procedure in [KIRG19] to compute the alignment given the similarity input matrix. The task is evaluated by intersection over detection (IoD), defined as $\frac{G \cap D}{D}$ the ratio between the intersection of ground-truth action G and prediction D to prediction D , and the Jaccard index, which is an (IoU) given as $\frac{G \cap D}{G \cup D}$.

9.5.4 Comparison to State-of-the-Art

(i) **Spatio-temporal grounding in untrimmed video:** We first compare the proposed method with other approaches designed for spatial or temporal grounding in Table 9.2. It shows that models without specific loss designs for spatial grounding (MIL-NCE [MAS⁺20], CLIP [RKH⁺21b]) show good mAP scores but lower pointing game accuracy. Out of the two weakly supervised methods, GLIP [LZZ⁺22] and RegionCLIP [YZY⁺22]), trained with aligned image-text, RegionCLIP show significantly better performance in this setting, while both perform in a similar range in the spatial grounding scenario (see Table 9.3). We attribute this behavior to the fact that RegionCLIP distinguishes frames with relevant queries better from background than GLIP, leading to better temporal localization. We finally compare the strong baseline MIL-NCE+RegionCLIP, which combines two approaches specialized in temporal and spatial aspects, to our task. It shows that the proposed method improves over all other baselines underlining the need to incorporate global (temporal) and local (spatial) representations.

Method	Backbone	Data	Super.	IoU	IoD
Mining: MLP [KIRG19]	TSM	MiningYT	Weak	9.80	19.20
CoMMA [TPS ⁺ 21]	S3D	HT250K	Self	2.05	5.63
RegionCLIP [ZYZ ⁺ 22]	RN50x4*	CC3M	Weak	10.96	16.86
MIL-NCE [MAS ⁺ 20]	S3D*	HT100M	Self	18.69	26.74
Ours	S3D	HT100M	Self	19.18	27.65
Ours	CLIP	HT100M	Self	19.88	28.50
Ours	CLIP*	HT100M	Self	20.33	29.67

Table 9.4: **Temporal Grounding on MiningYoutube.** Spatial-focused model CoMMA is not trained for temporal detection, which results in lower performance. In contrast, the proposed model combines global and local representation, resulting in better temporal localization than one alone. Super.=Supervision, * indicates finetuned backbone.

	GroundingYT Spatio-temporal	MiningYT Temporal	YC-Inter. Spatial	GroundingYT Spatio-temporal	MiningYT Temporal	YC-Inter. Spatial
	w/o Sinkhorn			with Sinkhorn		
None	17.43	18.34	57.42	-	-	-
Global T - Global V	18.24	19.38	56.31	18.96	19.89	57.34
Global T - Local V	18.01	19.31	57.56	18.53	19.35	57.67
Local T - Global V	18.05	19.85	57.34	19.32	20.36	58.13
Local T - Local V	18.31	19.48	57.86	19.43	20.16	58.51
Average over last two	18.36	19.68	57.77	19.45	20.33	58.35

Table 9.5: **Frame selection: (a)** Sinkhorn selection results in better supervision. **(b)** We investigate all possible combinations of global and local representations for frame selection similarity matching. We found keeping the local text representation is crucial, and a combination of local and global representation leads to the best spatio-temporal grounding result.

(ii) **Spatial grounding:** Second, we compare the performance of the proposed framework to other methods on the task of spatial grounding, including models with weak supervision, as well as models trained in a fully supervised setting in Table 9.3. In the instruction video domain (GYT and YC-Inter), the proposed approach achieves the best result among all weakly and self-supervised trained methods. In the general domain (V-HICO and Daly), the method also achieves competitive results, showing the generalizability of the model to other domains. Note that in the Daly dataset, the classes are verbs, which are not detectable by the object-focused model GLIP. Compared to their weakly trained counterparts, fully-supervised model (TubeDETER [YMS⁺22], STCAT [JyIYM22]) achieve competitive performance in the general domain (V-HICO, Daly) and slightly lower performance in instruction domain (GYT, YC-Inter) due to the domain gap with respect to the training data.

(iii) **Temporal grounding:** We evaluate temporal grounding in Table 9.4. Here, it shows that global representations also profit from local representation learning. This hypothesis is further validated in the ablation studies in Table 9.6, where we ablate both losses for all three settings and show a consistent improvement in the joint loss formulation.

9.5.5 Ablation Study

We perform ablation studies with respect to all three settings, spatio-temporal grounding, as well as spatial and temporal grounding alone, reporting performance for spatio-temporal

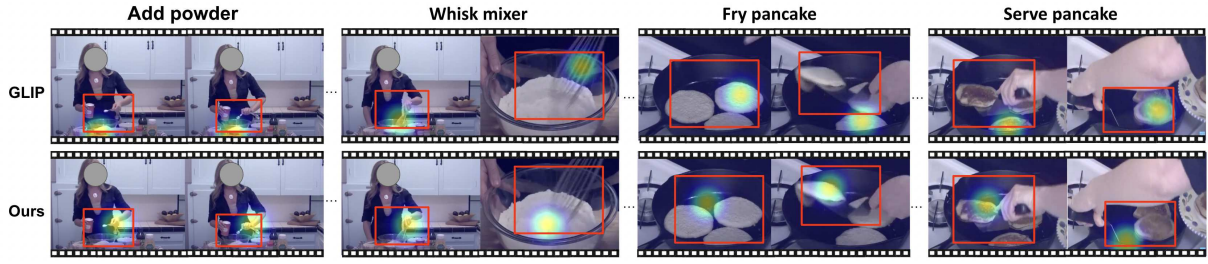


Figure 9.5: **Visualization on GroundingYoutube.** Red box: annotation. Heatmap: prediction. Baseline GLIP focuses on objects such as power, mixer, and pancake. Our method focuses on actions such as whisk and serve.

	GroundingYT Spatio-temp	MiningYT Temporal	YC-Inter. Spatial
only Local loss	7.29	5.23	55.29
only Global loss	9.28	19.12	36.23
w/ Both loss	19.45	20.33	58.35

Table 9.6: **Loss ablations:** both losses contribute to the final loss, and the existence of global loss helps the localization task.

grounding on GroundingYT using mAP with IoU@0.4, on temporal grounding using MiningYT IoU, and on spatial grounding using YC-Inter. pointing game. Additional ablations are in Appendix G.3.3.

Frame selection strategy. We perform an ablation on the possible frame selection strategies for our method (Figure 9.2(b) and Section 9.3.2). In Table 9.5, *None* uses all frames within the ASR boundary ($U = T$) as our video training data. *Global* represents the [CLS] token in text and video. *Local* uses the words and spatio-temporal tokens. In the setting Sinkhorn was not applied, the top T frames with the highest similarity score were selected. When we set spatio-temporal tokens as the selection target, we sum over the scores with respect to each frame to acquire the frame similarity score. It shows that selecting frames based on Sinkhorn selection leads to consistently better results as it enforces more variety of visual concepts but also captures frames with possible groundable objects. It further shows that word tokens are more suitable than the global text CLS token for frame selection. Finally, we see that depending on the task (spatial vs. temporal), a local resp. global representation is better, and a combination of both works best for spatio-temporal grounding. We provide runtime analysis of such frame selection strategy in Appendix G.3.1.

Global and local loss. As mentioned in the spatio-temporal evaluation, both features contribute to the final grounding result. We test the model by ablating out each loss. Table 9.6 shows that each loss not only contributes to the spatio-temporal grounding on the GYT, but also that the whole is more than the sum of its parts (losses) since this task requires both spatial and temporal detection. The reduced impact of the global loss in the case of YC-Inter is that this is a pure spatial grounding dataset (no background frames) without temporal detection, and the local loss plays a more critical role. We observe the same patterns in the temporal grounding result for MYT, where spatial localization is not directly contributing to the final performance. We tried out the same ablation using in the S3D backbone in Appendix G.3.

9.5.6 Qualitative Results

We visualize our spatio-temporal result in Figure 9.5. For the GLIP model, we output the bounding box with the highest confidence score and visualize its center point. We found GLIP model focuses on the salient object while our model focuses more on human-object interaction.

9.6 CONCLUSION

In this chapter, we presented an approach for learning spatio-temporal grounding solely from noisy multimodal supervision. In addition, we introduced the GroundingYoutube dataset, which provides dense spatio-temporal point and box annotations for untrimmed, multi-action videos. Our method includes a frame selection mechanism that identifies frames with groundable objects to adapt the learning process for untrimmed videos. Furthermore, we jointly learn global representations, which capture temporal information, and local representations learning fine-grained multimodal interactions between video and text. We conducted extensive experiments and our approach shows state-of-the-art performance in spatio-temporal grounding, as well as temporal and spatial grounding alone. In the next chapter, Chapter 10, we extend this research and show how the activity-grounding capacity of frozen pretrained multimodal models can be revealed without any additional training.

VIDEOGEM: TRAINING-FREE ACTION GROUNDING IN VIDEOS

Contents

10.1	Introduction	136
10.2	Related Work	138
10.2.1	Spatial Video Grounding	138
10.2.2	Training-free Vision-Language Grounding	138
10.3	Method	139
10.3.1	Background (GEM)	139
10.3.2	GEM for Videos	140
10.3.3	Layer Weighting	141
10.3.4	Prompt Decomposition for Action Grounding	142
10.4	Experimental Evaluation	143
10.4.1	Datasets	143
10.4.2	Implementation Details	144
10.4.3	Comparison to State-of-the-Art	144
10.4.4	Ablation Study	145
10.5	Conclusion	148

Image-language foundation models have shown impressive capabilities across various zero-shot tasks, including training-free localization and grounding, primarily focusing on localizing objects in images. However, leveraging those capabilities to localize actions and events in videos is challenging, as actions have less physical outline and are usually described by higher-level concepts. In this work, we propose VideoGEM, the first training-free spatial action grounding method based on pretrained image- and video-language backbones. Namely, we adapt the self-self attention formulation of GEM [BPFK24] to spatial activity grounding. We observe that high-level semantic concepts, such as actions, usually emerge in the higher layers of the image- and video-language models. We, therefore, propose a layer weighting in the self-attention path to prioritize higher layers. Additionally, we introduce a dynamic weighting method to automatically tune layer weights to capture each layer’s relevance to a specific prompt. Finally, we introduce a prompt decomposition, processing action, verb, and object prompts separately, resulting in a better spatial localization of actions. We evaluate the proposed approach on three image- and video-language backbones, CLIP, OpenCLIP, and ViCLIP, and on four video grounding datasets, V-HICO, DALY, YouCook-Interactions, and GroundingYouTube, showing that the proposed training-free approach is able to outperform current trained state-of-the-art approaches for spatial video grounding.

This chapter is based on the CVPR 2025 paper [VBK⁺25]. This paper is based on the master thesis of Felix Vogel. Nina Shvetsova acted as Felix Vogel’s supervisor during this project.

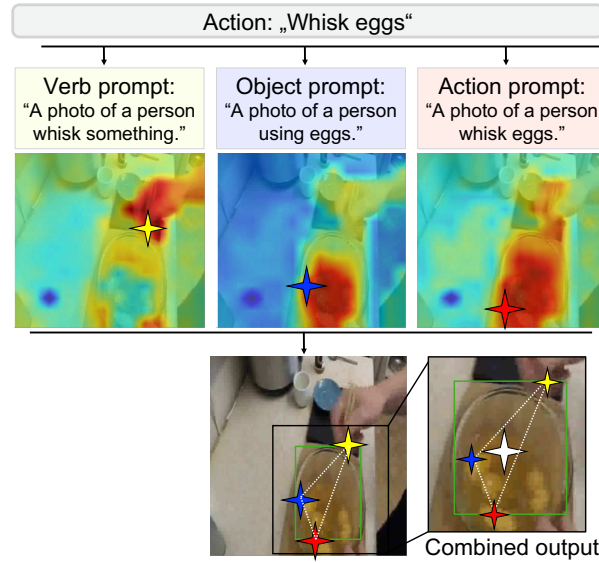


Figure 10.1: **Prompt decomposition and combination.** First, we decompose the action query into verb, object, and action prompts. For each component, we then predict locations corresponding to the highest values on the heatmaps (red-high, blue-low). To determine the final location (white star), we calculate the center point of these individual predictions. The red, blue, and yellow stars represent the predicted locations for the action, object, and verb prompts, respectively, while the dark green bounding box represents the annotated ground truth.

10.1 INTRODUCTION

Spatial localization of actions in videos has been a long-standing topic in video analysis [TZC⁺10, KMSZ12, GSR⁺18, MS17, SIS18, WMS16, SJL13, WHS15, SIS15]. While early approaches were trained with human-annotated bounding boxes and mainly focused on detecting humans and classifying respective actions [JGZ⁺13, SZS12], recent approaches [CSR⁺24, WNK⁺24, TPS⁺21], including our method from the previous chapter [CSR⁺24], are trying to leverage the localization properties of vision-language models without the need of bounding box annotations, learning localization only from image-text or video-text pairs. However, the localization of actions based on such weak supervision presents some significant challenges: unlike objects, which usually have a clear, unambiguous outline, actions often lack such region properties and exhibit diverse, context-dependent semantics. Additionally, actions often refer to interactions between entities and can comprise multiple objects or people over different timespans as shown in Figure 10.1. This variability makes it hard to capture the various visual representations of an action, especially based on web-crawled image- or video-text pairs. As a result, spatial video grounding methods [CSR⁺24, SXGX19, TPS⁺21] still need to be specifically trained, e.g. with a localization loss, to perform spatial localization.

Compared to that, another line of work focuses on training-free localization of semantic concepts in images [BPFK24, LWDL23, ZLD22] by proposing various training-free adaptations of vision-language models to perform object localization. Namely, CLIPSurgery [LWDL23] proposes an alternative pathway based on value-value attention which was extended by Grounding Everything Module (GEM) [BPFK24] to a self-self attention pathway. While those methods work well for object localization in image data, action localization requires models to capture contextual cues beyond object boundaries.

To address this problem, we introduce VideoGEM, the first training-free method for spatial activity grounding in videos. Inspired by GEM [BPFK24], which focuses on spatial object localization, VideoGEM extends this approach to spatially localize activities within video content. First, we extend the self-self attention pipeline to process multiple frames for action localization on video data for applying video backbones. In this setup, self-self attention automatically spans multiple frames, aggregating attention across both space and time. However, pretrained image-language but also video-language backbones struggle to capture abstract concepts such as actions. To address this problem, we analyze the self-self attention pipeline of several backbones, showing that abstract concepts such as actions and verbs usually arise in higher layers. We, therefore, propose to weight layers of the self-self attention pipeline based on a mixture of static and dynamic weights: while static weights give more weight to higher layers, dynamic weights are adjusted based on the relevance of a layer for each prompt.

We further observe that vision-language models often show a strong object bias when they are prompted with verb-object combinations [YBK⁺23b, BPFK24, VSKK22]. To counteract this and focus the model on both verb and object, we propose a prompt decomposition. To this end, the *verb* and *object* of the *action* description are extracted and separately prompted in addition to the original action. We then compute the center points of the resulting individual predictions for *verb*, *object*, and *action*, and consider the weighted mean of all three center point predictions (see Figure 10.1). This accounts for the fact that verbs might focus more on hands, while object heatmaps automatically capture the object.

We evaluate our approach on three pretrained backbones, CLIP [RKH⁺21b], OpenCLIP [SBV⁺22], and ViCLIP [WHL⁺23, XGH⁺21] and on four action grounding datasets, V-HICO [LDT⁺21], Daly [WMS16], YouCook-Interactions [TPS⁺21], and GroundingYouTube [CSR⁺24]. It shows that VideoGEM allows image and video backbones to spatially ground actions in a training-free zero-shot manner and that they are able to outperform models specially trained for this setup. We further analyze the impact of the proposed components, including the effect of video vs image processing for self-self attention as well as the impact of layer weighting and prompt decomposition, showing how those factors contribute to the final performance.

We summarize our contributions as follows.

- We propose VideoGEM, the first training-free method for spatial action grounding in videos that adapts self-self attention to the video domain.
- To capture higher-level semantic concepts such as actions, we propose a mixture of static and dynamic weights that prioritizes layers according to their relevance for capturing such concepts.
- We propose prompt decomposition to address the object bias in vision-language models, allowing self-self attention to consider actions, verbs, and objects independently.
- We show that VideoGEM outperforms even finetuned localization methods and provide extensive analysis of all the components.

10.2 RELATED WORK

10.2.1 Spatial Video Grounding

Spatially localizing actions in videos without the need to explicitly label respective instances has drawn significant attention in the last years. As one of the first works, CoMMA (Contrastive Multilayer Multimodal Attention) [TPS⁺21] proposes a multilayer cross-modal attention network that obtains an attention heatmap via attention rollout, where the predicted location is the maximum attention. Moreover, the authors propose the YouCook-Instructions dataset, based on YouCook2 [ZXC18], to evaluate spatial grounding for models pretrained on YouTube cooking data. Compared to that, TubeDETR [YMS⁺22] uses a space-time decoder, decoding the video-text features into a spatiotemporal object tube which includes box annotations per frame. STCAT [JylYM22] proposes a Spatio-Temporal Consistency-Aware Transformer that also uses a vision and text encoder, followed by cross-model interaction. To create the object tube, the authors generate multi-modal templates to guide the decoder, followed by a prediction head. Recently, VideoGrounding-DINO [WNK⁺24] extends GroundingDINO [LZR⁺23] to videos. Finally, What-When-Where [CSR⁺24] proposes a global and local loss together with a frame selection mechanism to detect actions in untrimmed videos in space and time. The authors also evaluate on a new benchmark, GroundingYouTube, which is based on MiningYouTube [KIRG19], to evaluate spatiotemporal grounding in untrimmed videos, and also use the annotated segments to evaluate spatial grounding alone. In contrast, our method is training-free, using the original backbone weights without additional finetuning, whereas the previous methods rely on additional training of either a new projection head or the finetuning of the full model.

10.2.2 Training-free Vision-Language Grounding

The success of large-scale vision-language models like CLIP has generated significant interest in applying them to tasks such as open-vocabulary object localization. In this context, a special line of methods focuses on training-free localization [ZLD22, LWDL23, BPFK24], thus adapting pretrained vision-language models to handle localization tasks without changing the weights of the pretrained model. MaskCLIP [ZLD22] achieves this by removing the Multi-Layer Perceptron (MLP) in the vision transformer’s final layer, using the last value projection to capture dense patch-level features. Building on this, CLIPSurgery [LWDL23] introduces a parallel “surgery pathway” alongside the standard CLIP vision transformer (ViT) backbone, which operates with value-value attention instead of the usual query-key attention and connects outputs from multiple layers through residual connections. Consistent with MaskCLIP [ZLD22], CLIPSurgery omits the final MLP, directly applying value-value attention. GEM [BPFK24] further improves the concept of value-value attention by generalizing it to self-self attention, not only using value-value attention but also query-query and key-key attention together with a set of regularizations. We extend the GEM model and the self-self attention mechanism by adapting it to video input and introducing a weighting and a prompt decomposition mechanism, applying it to action localization.

10.3 METHOD

In the following, we first review the GEM self-self attention pipeline in Section 10.3.1. We then discuss its extension to videos in Section 10.3.2, the proposed layer weighting in Section 10.3.3, and the prompt decomposition in Section 10.3.4.

10.3.1 Background (GEM)

In the original ViT paper [DBK⁺20], the attention operation is computed as follows:

$$\begin{aligned} \mathbf{A} &= \text{softmax} \left(\frac{\mathbf{X}\mathbf{W}_q(\mathbf{X}\mathbf{W}_k)^T}{\tau} \right), \\ \mathbf{O} &= \mathbf{A} \cdot (\mathbf{X} \cdot \mathbf{W}_v), \end{aligned} \quad (10.1)$$

where $\mathbf{X} = (x_i)_{i \leq N} \in \mathbb{R}^{N \times d}$ represents the patch tokens output from the ViT, N is the number of visual tokens, d is the dimension of each token and $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d \times (h \cdot d_h)}$ are the projection matrices, where h is the number of heads in the attention and d_h is the dimension of each head. The Grounding Everything Module (GEM) [BPFK24] introduces a parallel pathway that operates alongside the original trained ViT while sharing its weights. This pathway replaces the standard self-attention mechanism with a self-self attention operation defined as follows:

$$\mathbf{A}_{ss} = \text{softmax} \left(\frac{(\mathbf{X}\mathbf{W}_{proj})(\mathbf{X}\mathbf{W}_{proj})^T}{\tau} \right), \quad (10.2)$$

where $\mathbf{W}_{proj} \in \{\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v\}$, and τ is the temperature. The self-self attention is applied iteratively J times on L^2 normalized visual tokens. For input visual tokens $\mathbf{X} \in \mathbb{R}^{N \times d}$, we denote $\mathbf{P}^{(j)}$ as the output of the self-self attention at iteration j :

$$\begin{cases} \mathbf{P}^{(0)} = \frac{\mathbf{X}\mathbf{W}_{proj}}{\|\mathbf{X}\mathbf{W}_{proj}\|_2}, \\ \tilde{\mathbf{P}}^{(j)} = \text{softmax} \left(\frac{\mathbf{P}^{(j-1)}(\mathbf{P}^{(j-1)})^T}{\tau} \right) \mathbf{P}^{(j-1)}, \\ \mathbf{P}^{(j)} = \frac{\tilde{\mathbf{P}}^{(j)}}{\|\tilde{\mathbf{P}}^{(j)}\|_2}. \end{cases} \quad (10.3)$$

The final output $\mathbf{O}_{ss}$ is obtained by applying the attention matrix to the values:

$$\mathbf{O}_{ss} = \text{softmax} \left(\frac{\mathbf{P}^{(J)} \cdot (\mathbf{P}^{(J)})^T}{\tau} \right) \cdot V. \quad (10.4)$$

The final self-self attention output is obtained by averaging the query-query, key-key, and value-value attention:

$$\mathbf{O}_{qkv} = \frac{(\mathbf{O}_{qq} + \mathbf{O}_{kk} + \mathbf{O}_{vv})}{3}. \quad (10.5)$$

This formulation enables the patch tokens alignment with the Vision-Language Model (VLM) text encoder. Localization heatmaps can then be constructed by computing the cosine similarity between the text embedding of a prompt and the corresponding patch token representations.

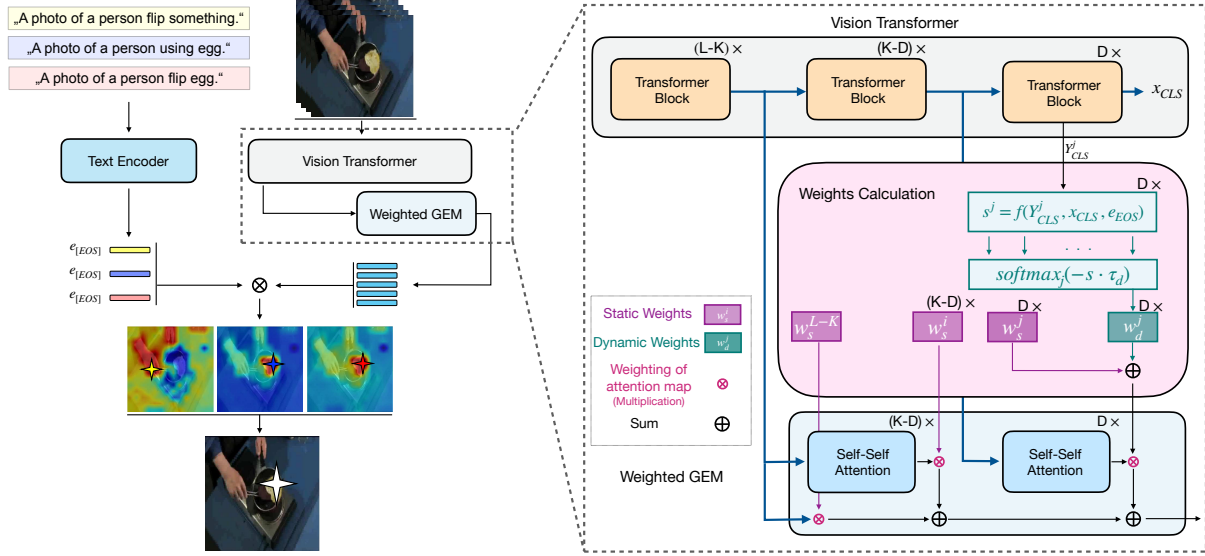


Figure 10.2: **Left: VideoGEM pipeline.** VideoGEM takes a video and its corresponding narration as input. Our *Weighted GEM* processes the input video alongside the vision transformer to generate the representative patch tokens. Decomposition of the input narration into verb prompt, object prompt, and action prompt (see Section 10.3.4 for details) are passed through the text encoder to obtain three $[EOS]$ tokens, respectively. Then, three heatmaps are calculated as a similarity between patch tokens and the respective $[EOS]$ tokens. We then aggregate the heatmaps into one final prediction by centering the individual predicted locations. **Right: Layer weighting.** In our *Weighted GEM* architecture, we apply a combination of static and dynamic weights. Dynamic weights are applied to the last D layers, while static weights are applied to the last K layers, with $K > D$. Additionally, the attention map X^{L-K} is weighted by a corresponding static weight w_s^{L-K} . All weighted outputs from the self-attention blocks are then summed with the weighted X^{L-K} attention map to produce representative patch tokens. The output patch tokens of weighted GEM are used for similarity calculation with the text, resulting in an attention heatmap.

10.3.2 GEM for Videos

We extend GEM to handle video inputs by adapting it to video-language models, using the ViCLIP backbone as an example. Given a video input, ViCLIP processes a sequence of $T = 8$ frames $\mathbf{F} = \{f_1, \dots, f_T\}$. Each frame f_i is divided into N patches, resulting in a total of $T \times N$ patch tokens. These tokens are processed jointly across all frames through the transformer layers, allowing the model to capture both spatial and temporal relationships.

For a given input sequence, the patch tokens from all frames are concatenated into a single sequence $\mathbf{X} \in \mathbb{R}^{(T \times N) \times d}$, where d is the embedding dimension. The self-self attention mechanism is then applied to this combined sequence of tokens. After that, we compute the cosine similarity between each patch token and the prompt embedding of the text encoder, resulting in a similarity score matrix $\mathbf{S} \in \mathbb{R}^{(T \times N)}$. To generate the final localization output, we focus on a target frame f_t positioned such that we maintain temporal context from both past and future frames. The similarity scores for frame f_t are reshaped and interpolated to match the spatial dimensions of the input frame and min-max normalized to produce the final attention heatmap. Note, that the temporal information of the video is solely considered by the

video backbone.

10.3.3 Layer Weighting

In the original GEM formulation, the self-self attention outputs are combined through residual connections, effectively giving equal importance to all layers in the final prediction. However, we observe that higher-level concepts, particularly actions, predominantly emerge in the higher layers of the network. Based on this, we propose a combination of static and dynamic weights to prioritize the contributions of different layers. We extend the definition of $\mathbf{X}$ from the previous section, to $\mathbf{X}^l \in \mathbb{R}^{(T*N) \times d}$ which represents the l -th transformer block output, where $l \in \{1, \dots, L\}$ and L is the total number of transformer blocks. Let $\mathbf{Y}^l \in \mathbb{R}^{(T*N) \times d}$ denote the output of the l -th transformer block before the residual connection. Similarly, let $\mathbf{Z}^l \in \mathbb{R}^{(T*N) \times d}$ represent the output of the parallel self-self attention pathway at layer l before the residual connection. Then, when applied to the last K layers of the ViT, the output of GEM can be reformulated as follows:

$$\mathbf{O}_{GEM} = \mathbf{X}^{L-K} + \sum_{l=L-K+1}^L \mathbf{Z}^l. \quad (10.6)$$

Static Weighting. We first introduce static weights to assign specific importance to individual layers. For layer l , the static weight is defined as w_s^l for $l \in \{L-K, \dots, L\}$ resulting in the new output:

$$\mathbf{O}_{stat} = w_s^{L-K} \cdot \mathbf{X}^{L-K} + \sum_{l=L-K+1}^L w_s^l \cdot \mathbf{Z}^l. \quad (10.7)$$

Practically, w_s^l increases monotonically with l .

Dynamic Weighting. We further introduce dynamic weights that adapt to the semantic requirements of each input prompt by analyzing each layer's contribution to the model's understanding of the prompted concept.

Let $\mathbf{x}_{CLS} = \mathbf{X}_0^L \in \mathbb{R}^d$ denote the [CLS] token representation from the final layer, which can be decomposed into the sum of residuals from all layers: $\mathbf{x}_{CLS} = \sum_{l=1}^L \mathbf{Y}_0^l = \sum_{l=1}^L \mathbf{Y}_{CLS}^l$. The text embedding of the prompt obtained from the text encoder is defined as $\mathbf{e}_{EOS} \in \mathbb{R}^d$. We measure each layer's importance by evaluating how its removal affects the alignment between visual and textual representations, for details see Section 10.4.4. Let $s = (s^{L-D+1}, \dots, s^L)$ be the similarity vector where we compute the similarity score for layer l as:

$$s^l = \frac{(\mathbf{x}_{CLS} - \mathbf{Y}_{CLS}^l) \cdot \mathbf{e}_{EOS}}{\|\mathbf{x}_{CLS} - \mathbf{Y}_{CLS}^l\|_2 \cdot \|\mathbf{e}_{EOS}\|_2}, \quad (10.8)$$

The dynamic weights for the last D layers with $D \leq K$ are computed using the similarity vector s through a softmax operation with temperature τ_d . We define the dynamic weight for layer $l \in \{L-D+1, \dots, L\}$ as:

$$w_d^l = \text{softmax}_l(-s \cdot \tau_d), \quad (10.9)$$

then the final output for dynamic weights is computed as:

$$\mathbf{O}_{dyn} = \mathbf{X}^{L-K} + \sum_{l=L-K+1}^{L-D} \mathbf{Z}^l + \sum_{l=L-D+1}^L w_d^l \cdot \mathbf{Z}^l. \quad (10.10)$$

It is important to note that the [CLS] token representations used for computing similarities are extracted from the original self-attention pathway rather than the self-self attention. It is motivated by the fact that the [CLS] token in the original model was specifically trained to align with the text encoder's [EOS] token representations during pretraining.

Combining Static and Dynamic Weights. To leverage the complementary benefits of both weighting schemes, we propose a unified approach that combines static and dynamic weights (see Figure 10.2). For a network with L layers, we apply static weights to the last K self-self attention layers and the previous self attention input while computing dynamic weights for the last D layers (where $D \leq K$). The combined weights w_c^l for $l \in \{L - K, \dots, L\}$ are defined as:

$$w_c^l = \begin{cases} w_s^l - \frac{1}{D} + w_d^l & \text{if } l > L - D \\ w_s^l & \text{otherwise} \end{cases} \quad (10.11)$$

where w_s^l denotes static weight and w_d^l represents the dynamic weight as defined in Equation (10.9). We subtract $\frac{1}{D}$ from static weights when dynamic weights are applied to not increase the sum of weights. The final output incorporating these combined weights is computed as:

$$\mathbf{O}_{comb} = w_c^{L-K} \cdot \mathbf{X}^{L-K} + \sum_{l=L-K+1}^L w_c^l \cdot \mathbf{Z}^l \quad (10.12)$$

This weighted combination allows the model to adaptively balance the static prior knowledge about layer importance with dynamic, prompt-specific adjustments, particularly in higher layers where semantic concepts emerge.

10.3.4 Prompt Decomposition for Action Grounding

Action descriptions typically consist of two key components: A verb describing the action itself and one or more objects involved in the action. To effectively leverage this inherent structure, we propose a prompt decomposition method that processes these components separately while maintaining the context of the complete action description. Namely, we decompose each action query into three distinct components: a verb prompt, an object prompt, and the original action prompt. For consistent processing, we employ the following template format:

Prompt Templates

- Verb: *A photo of a person {verb} something.*
- Object: *A photo of a person using {object}.*
- Action: *A photo of a person {action prompt}.*

In case a component, verb or object, is missing or cannot be extracted, we use fallback templates, "*A photo of a person doing something.*" for missing verbs and "*A photo of a person.*" for missing objects. For each prompt, we compute the similarity between its text embedding and the visual tokens, identifying the regions of highest activation. The model then determines a center point for each attention map, corresponding to the location with maximum similarity between the text and visual representations.

To combine these separate predictions into a final localization output, we employ a weighted averaging scheme that prioritizes the action prompt while incorporating information from the component-specific predictions (see Figures 10.1 and 10.2). Let $c_{verb}, c_{obj}, c_{act} \in \mathcal{R}^2$ denote the predicted center coordinates for the verb, object, and action prompts respectively, and $w_{verb}, w_{obj}, w_{act} \in \mathcal{R}$ their corresponding weights, where we assign higher weights to action prompts. The final prediction c_{dec} is computed as:

$$c_{dec} = w_{verb} \cdot c_{verb} + w_{obj} \cdot c_{obj} + w_{act} \cdot c_{act} \quad (10.13)$$

Model	Backbone	Backbone Pretraining Data	Grounding Training Data	VH	Daly	YC	gYT	Avg.
Models trained for grounding with localization supervision:								
TubeDETR [†] [YMS ⁺ 22]	ResNet101 [HZRS16], RoBERTa [Liu19]	ImageNet, Visual Genome, Flickr, COCO	VidSTG	-	-	51.63	-	-
STCAT [†] [JylyM22]	ResNet101 [HZRS16], RoBERTa [Liu19]	-	VidSTG	-	-	55.90	-	-
VideoGrounding-DINO [WNK ⁺ 24]	Swin-L; BERT [DCLT18]	O365, OI, GoldG, Cap4M, COCO, RefC	VidSTG	-	-	57.73	-	-
GLIP [‡] [LZZ ⁺ 22]	Swin-L* [LLC ⁺ 21]	-	FourODs,GoldG,Cap24M	66.05	-	52.84	53.62	-
Models trained for grounding with vision-text pairs only:								
CoMMA [‡] [TPS ⁺ 21]	CLIP	WIT-400M	HT100M	55.20	61.06	52.65	47.56	54.12
RegionCLIP [‡] [ZYZ ⁺ 22]	RN50x4*	WIT-400M	CC3M	57.92	67.12	51.56	52.84	57.36
WWW-CLIP [CSR ⁺ 24]	CLIP	WIT-400M	HT100M	60.71	70.08	57.10	55.49	60.85
WWW-CLIP [CSR ⁺ 24]	CLIP*	WIT-400M	HT100M	62.34	71.35	58.35	56.98	62.26
No training for grounding:								
GEM [BPFK24]	CLIP	WIT-400M	-	67.79	78.52	50.08	36.92	58.33
	OpenCLIP	LAION2B	-	68.28	74.05	56.87	32.91	58.03
	ViCLIP	InternVid-FLT-10M	-	65.08	73.75	53.62	51.28	60.93
VideoGEM (ours)	CLIP	WIT-400M	-	76.90	84.53	52.57	47.46	65.37
	OpenCLIP	LAION2B	-	76.42	80.32	60.05	45.33	65.53
	ViCLIP	InternVid-FLT-10M	-	75.75	78.25	55.10	57.21	66.58

Table 10.1: **Accuracy of VideoGEM compared to the State-of-the-Art.** VideoGEM includes prompt decomposition, and static and dynamic weights. GEM is applied with the same action prompt as VideoGEM. We compare the accuracy on V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT). Finetuned backbones are marked with *, [†] denotes results from [WNK⁺24], [‡] denotes results from [CSR⁺24].

The weighted combination serves two purposes. First, it maintains the primacy of the action prompt’s prediction while allowing refinement based on component-specific localizations. Second, if any single component prediction deviates from the true action center, the ensemble nature of the prediction helps maintain accuracy through the influence of the other components. This is particularly valuable for complex actions where the verb and object locations might provide complementary spatial information.

Extracting Verbs and Objects. For datasets with a label structure of “*verb_object*” verb and object can be directly retrieved. For natural language annotations as in YouCook-Interactions, we select the verbs and objects as the ones that are most likely visible in the input. To this end, we extract all verbs and objects from the action description with a natural language processing (NLP) tool. We then generate an individual prompt for each verb and object and apply the respective vision-language backbone to determine which verb and object show the highest similarity to the input.

10.4 EXPERIMENTAL EVALUATION

10.4.1 Datasets

We evaluate four video datasets for action grounding. Since our proposed method is training-free, we only consider the test set of the datasets for our evaluation, if there is one.

V-HICO (Videos of Humans Interacting with Common Objects) [LDT⁺21] has a test set containing 608 videos with annotated bounding boxes for a human performing an action and

the object on which the action is performed. It consists of 244 object classes and 99 action classes with a total of 756 action-object pairwise classes. Following the evaluation of [CSR⁺24] we use the union of the human and object bounding boxes as ground truth for V-HICO.

DALY (Daily Action Localization in YouTube videos) [WMS16] annotates 510 YouTube videos with ten action classes.

GroundingYouTube [CSR⁺24] is based on the MiningYouTube dataset [KIRG19] which contains 250 cooking videos from YouTube. GroundingYouTube provides spatio-temporal annotations including bounding boxes for actions.

YouCook-Interactions [TPS⁺21] is based on the validation set of YouCook2 [ZXC18] that contains 457 videos. It provides spatial bounding box annotations for interactions with labels as natural language sentences.

10.4.2 Implementation Details

We use GEM with $K = 7$ self-self attention layers and $J = 1$ iterations according to GEM [BPFK24], using the static weights: $w_s = [0.3, 0.4, 0.5, 0.6, 0.7, 0.9, 0.9, 0.9]$ for the seven self-self attention layers, and its initial self attention input. Note that w_s starts at index $L - K$. We further apply dynamic weights for the last $D = 3$ layers. We use a temperature of $\tau_d = 20$ for dynamic weights according to Equation 10.9 for all datasets and models. For prompt decomposition we apply the weights $w_{verb} = 0.2, w_{obj} = 0.2, w_{act} = 0.6$ according to Equation 10.13. We evaluate on ViCLIP, OpenCLIP, and CLIP as backbones in a training-free manner meaning that they are not specifically trained for action localization. For ViCLIP, we sample 7 frames around the labeled frame to get a video input. 4 frames are sampled before, 3 frames are sampled after the labeled frame. Accuracy is used as the main evaluation metric. A prediction is correct if the predicted location is inside the ground-truth bounding box, otherwise it is false. The accuracy is calculated as the proportion of the correct predictions.

10.4.3 Comparison to State-of-the-Art

We first compare the proposed approach to trainable video grounding methods as well as to the vanilla GEM approach in Table 10.1. To the best of our knowledge, all existing video grounding models are either specifically trained for action grounding or have a finetuned backbone for action localization, whereas our setup is training-free.

First, we show that on average our VideoGEM outperforms all other methods by more than 3% with any of the three backbones. Looking at the different backbones in detail, we notice that especially CLIP and OpenCLIP, so backbones specialized for objects, perform well on the V-HICO and Daly datasets, while ViCLIP significantly outperforms all other backbones on GroundingYouTube. We attribute this to the fact that V-HICO and Daly focus mainly on object annotations, while GroundingYouTube mainly focuses on annotations based on the center point of the action, therefore, stronger deviates from object-centered bounding boxes. Moreover, YouCook-Interactions and GroundingYouTube only contain videos from the specialized cooking domain, while V-HICO and Daly contain more general actions. General actions might be easier recognizable in a zero-shot object-centric setting without specific domain knowledge. Compared to the original GEM pipeline, we show that VideoGEM always improves over GEM independently of the dataset and the backbone that is chosen.

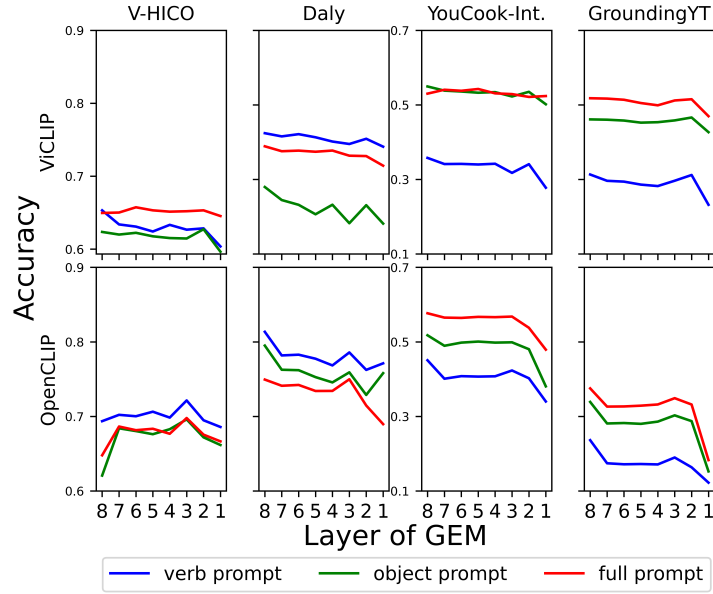


Figure 10.3: **Importance of GEM layers.** The accuracy of GEM with one removed layer is calculated. The removed layer index is on the x-axis where 1 is the final layer of GEM going down to 8 which is the initial self attention input to GEM.

10.4.4 Ablation Study

Layer Importance. Figure 10.3 we first visualize the per-layer impact by excluding single layers from one to eight layer from GEM. We observe that the accuracy is lower when the final layers of GEM are excluded compared to excluding early layers. This might lead to the assumption, that reducing the number of GEM layers would boost performance as well. Compared to that however, Figure 10.4 shows the results when adding up to seven layers with zero layers corresponding to the output of the backbone without GEM. It shows that the performance of GEM depends on the overall number of self-self attention layers and that the accuracy increases for more layers saturating at a depth of seven that we use for all our experiments, also suggested by the original GEM paper [BPFK24]. This strengthens the concept of static weights, suggesting that higher layers generally capture complex concepts better than earlier layers, justifying why they should be weighted higher in the final decision. Earlier layers on the other hand can still capture important concepts such that they should not be excluded entirely, but just given less importance for the final decision.

Layer Weighting. To assess the effect of different weighting strategies, we compare static, dynamic, and combined weights (Equation 10.7, 10.10, and 10.12) with using no weights in Table 10.2 observing a performance increase by about 1% on average for both backbones compared to no layer weighting. We observe that the effect of dynamic weighting depends on the backbone as it relies on representative *[CLS]* tokens for the final layers. If the *[CLS]* token is primarily formed in the last layer, the benefit of dynamic weights is reduced, as seen with the ViCLIP backbone. However, with OpenCLIP, we observe consistent improvements. Notably, on the Grounding YouTube dataset as the most distinct from other object-centric datasets, dynamic weights boost performance by more than 3%. We provide further insights in Appendix H.2.

Prompt Decomposition. We evaluate the effect of prompt decomposition in Table 10.3 by applying VideoGEM without prompt decomposition for only a verb, object, or action prompt,

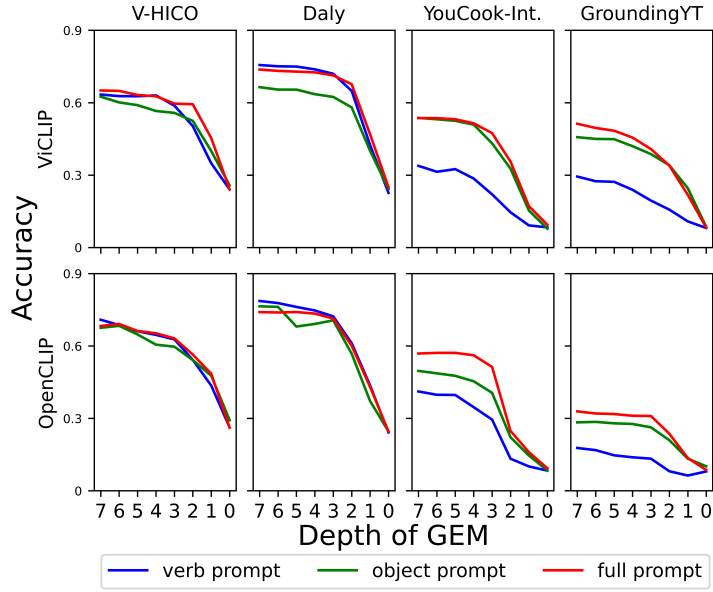


Figure 10.4: **Influence of the number of GEM layers.** Up to seven layers are added for GEM starting with a self-self attention layer for the final Transformer block. With zero layers, the output equals to the output of the backbone without GEM.

compared to our proposed prompt decomposition method including all prompts. Averaged over all datasets, using prompt decomposition improves over using only a single (verb, object, or action-) prompt by over 5% independently of the used backbone. Each prompt alone achieves strong performance, supporting the approach of first independently processing prompts and only then aggregating their individual predictions. Our method demonstrates their compatibility for recognizing complex activities effectively.

Additionally, we compare different merging strategies in Table 10.4 for obtaining the final prediction given the three attention heatmaps for the verb, object, and action-prompt. VideoGEM that averages the predicted positions, is evaluated against averaging or multiplying heatmaps elementwise. The ratio of weights for the verb, object, and action prompt is always 1 : 1 : 3 independent of the merging strategy. Compared to standard GEM, all merging strategies improve significantly. That suggests that decomposing the prompt into its relevant parts enforces the model to focus on every important part compared to only using the full prompt where the model can neglect the verb and mainly focus on the object [YBK⁺23b, BPFK24, VSKK22]. Moreover, averaging positions boosts performance much further compared to merging heatmaps (additively, or multiplicatively). This can be explained by two effects. First, it centers the action by predicting a position between the main parts of the action making it more robust. Second, it has self-correcting abilities. If one prediction is slightly off, the other predictions can drag the wrong prediction back to the others. It can thus be seen as an ensemble model having three votes for the correct prediction instead of just one, improving robustness and accuracy.

Image vs. Video Data. To determine the importance of video data for action grounding, we compare ViCLIP for video and image input in Table 10.5. ViCLIP takes eight images as input. In the image setting, we give the same image repeated eight times as an input. For the video setting, we use subsequent frames as input according to Section 10.4.2. Using video input outperforms its image-based counterpart independently of using standard GEM or VideoGEM

Backbone	Weights	VH	Daly	YC	gYT	avg
ViCLIP	none	74.79	76.84	54.38	56.39	65.60
	dyn	74.49	76.85	54.62	56.47	65.61
	stat	76.18	78.38	55.02	56.75	66.58
	s+d	75.75	78.25	55.10	57.21	66.58
OpenCLIP	none	77.86	79.27	59.20	40.17	64.13
	dyn	78.41	79.07	59.77	43.29	65.14
	stat	76.12	80.30	61.82	42.96	65.30
	s+d	76.42	80.32	60.05	45.33	65.53

Table 10.2: **Influence of different layer weighting strategies.** VideoGEM with prompt decomposition is applied without layer weighting (none), only static (stat), only dynamic (dyn), or combined weights (s+d) on V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT).

Backbone	Mode	VH	Daly	YC	gYT	avg
ViCLIP	verb	64.72	76.21	36.54	31.26	52.18
	obj	62.30	68.71	54.94	46.78	58.18
	act	65.68	74.17	52.97	51.99	60.95
	all	75.75	78.25	55.10	57.21	66.58
OpenCLIP	verb	69.60	80.24	43.45	21.60	53.72
	obj	65.44	77.00	51.21	34.20	56.96
	act	66.77	74.87	57.36	38.38	59.35
	all	76.42	80.32	60.05	45.33	65.53

Table 10.3: **Influence of prompt decomposition.** VideoGEM with static and dynamic weights is applied without prompt decomposition to only verb, object (obj), or action (act) prompts. The results are compared to VideoGEM with static and dynamic weights as well as prompt decomposition (dec) on V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT).

by almost 1% on average. While the video input increases accuracy on YouCook-Interactions and GroundingYouTube compared to the image input, it performs similarly on V-HICO and Daly. This can be attributed to the more static actions in V-HICO and Daly like *"snapping fingers"* for V-HICO or *"Drinking"* for Daly, while f.e. *"chopping"* or *"arranging"* in cooking videos as in YouCook-Interactions or GroundingYouTube is more dynamic. Note, that the only difference between the video and image input is, that the image input repeats the same image 8 times, while for video input surrounding frames are used. If the actions are static and the surrounding frames are very similar, the video and image input are also very similar resulting in only minor performance differences.

Backbone	Mode	VH	Daly	YC	gYT	avg
ViCLIP	none	65.68	74.17	52.97	51.99	60.95
	mul	66.16	76.72	55.91	48.87	61.92
	avg	65.44	75.99	56.75	49.96	62.04
	ours	75.75	78.25	55.10	57.21	66.58
OpenCLIP	none	66.77	74.87	57.36	38.38	59.35
	mul	68.28	80.36	58.40	34.18	60.31
	avg	68.40	78.94	58.04	35.34	60.18
	ours	76.42	80.32	60.05	45.33	65.53

Table 10.4: **Influence of merging strategies for prompt decomposition.** VideoGEM (ours) combines verb, object, and action prompt predictions with a weighted average of the predicted positions. This combination technique is compared to the element-wise multiplication (mul) and element-wise averaging (avg) of heatmaps before taking the highest attention value of the resulting heatmap as prediction. *None* corresponds to the baseline evaluation without prompt decomposition for action prompts with combined weights. We use V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT) for testing.

Model	Data	VH	Daly	YC	gYT	avg
GEM	vid	65.08	73.75	53.62	51.28	60.93
	img	65.20	74.00	52.17	48.80	60.04
VideoGEM	vid	75.75	78.25	55.10	57.21	66.58
	img	74.19	78.47	54.86	55.08	65.65

Table 10.5: **Influence of video input.** VideoGEM with ViCLIP as backbone is evaluated on image and video inputs. For reference we also evaluate GEM with ViCLIP as a backbone. We evaluate on V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT).

10.5 CONCLUSION

In this chapter, we introduced VideoGEM, the first training-free method for spatial action grounding in videos. We proposed a weighting technique using static and dynamic weights to assign greater importance to layers that capture conceptually relevant information for complex activities. Additionally, we introduced prompt decomposition to fully leverage action prompts, helping to reduce object bias in standard image- and video-language models. Remarkably, our training-free VideoGEM outperforms all previous state-of-the-art methods that rely on finetuning backbones for spatial action localization tasks.

It is important to note that, in this chapter, we focus on spatial grounding in a training-free setting, temporal grounding is not addressed. We leave the training-free spatio-temporal extension for future work.

In this part, Part IV, we considered two methods for activity grounding in video without any human annotations. This part concludes the main body of the thesis. In the next section, Chapter 11, we discuss the main conclusions of the thesis and outline directions for future work.

CONCLUSION AND FUTURE WORK

Contents

11.1	Key Ideas and Insights	150
11.2	Summary of Key Conclusions	150
11.2.1	Pretraining Representations Without Human Supervision	151
11.2.2	Finetuning Multimodal Models Without Human Supervision	151
11.2.3	Improving Data Quality Without Human Supervision	152
11.2.4	Grounding with Multimodal Models Without Human Supervision	152
11.3	Future Directions	153
11.3.1	Data: Building Robust Pretraining and Evaluation Corpora	153
11.3.2	Models: Learning Efficient Multimodal Representations	155
11.3.3	Analysis and Adaptation of Learned Representations	157
11.3.4	A Broader View on the Topic	158

In this thesis, we explored how to build robust computer vision models for image and video understanding without relying on human annotations, as human-labeled datasets are costly to create at scale, often contain noise and systematic biases [TE11], and follow annotation protocols shaped by human conventions [BSoo]. **Training without human annotations** opens the door to learning directly from raw signals, leveraging the natural structure of data. This not only enables scaling beyond the limits of human labels but also fosters models that are more flexible, less biased, and better aligned with the richness of real-world data. However, doing so raises several challenges across the model lifecycle. This thesis examined four such stages:

- (I) **Pretraining representations without human supervision** (Part I). We studied how to learn strong unimodal and fused multimodal representations from unlabeled images and naturally co-occurring web signals (video–audio–text). We explored contrastive objectives for learning coherent embedding neighborhood and fusion architectures that support any combination of input modalities and enable robust multimodal fusion.
- (II) **Finetuning multimodal models without human supervision** (Part II). Large pretrained vision-language models offer strong zero-shot performance, yet still lag behind models trained with full supervision. In this thesis, we took first steps toward adapting vision-language models to downstream tasks/domains without paired supervision using only unlabeled vision data and/or downstream language data.
- (III) **Improving data quality without human supervision** (Part III). Model robustness is constrained by the quality of the data used for training and evaluation. We developed methods to automatically clean and enrich training data at scale (e.g., denoising noisy web video narrations) and to assess and mitigate biases in evaluation benchmarks, without manual labeling.
- (IV) **Grounding with multimodal models without human supervision** (Part IV). Dense semantic tasks are essential because they specify *where* and *when* evidence occurs. We explored spatio-temporal and spatial grounding of actions in videos without bounding-box or moment annotations, using weak, freely available multimodal signals.

This chapter discusses the key ideas developed in this thesis in Section 11.1, summarizes main conclusions and the connections across chapters in Section 11.2, and reflects on future research directions and the broader view of this topic in Section 11.3.

11.1 KEY IDEAS AND INSIGHTS

We addressed the overarching goal of building robust vision systems without human annotations, from the data perspective through to dense tasks, by exploiting *noisy, web-scale multimodal supervision* and *language as a reasoning space*, as well as *advanced learning objectives, architectures, and pipelines*. In the following, we present the **key ideas and insights** developed in this thesis in more detail.

- (I) **Web multimodal data beyond modality alignment.** We found that naturally co-occurring web signals (e.g., video–audio–text) have a supervision not only to train unimodal encoders with contrastive objectives, as in prior work, but also to learn *modality fusion* (Chapter 4) and enable models with *dense* capabilities such as spatio–temporal grounding or spatial grounding (Chapter 9 and 10). Moreover, generalized multimodal models, e.g., CLIP [RKH⁺21b], also serve as *effective pseudo-supervision* for domain- or task-specific adaptation (Chapter 5 and 6), which allows finetuning for a particular task or domain without paired human annotations.
- (II) **Language as an operating space.** We found that language is a powerful intermediate space for reasoning about vision. Zero-shot LLMs can process automatically recognized speech from long videos and produce high-quality human-written-like descriptions without even seeing the visual data (Chapter 7). Isolating specific concepts in text is also easier than in a video/image, that combined with strong LLM reasoning abilities supports systematic analysis of representation biases of visual dataset purely within the language space (Chapter 8). We believe that reasoning in the language space can provide a powerful interface for developing vision models.
- (III) **Objectives, architectures, and pipeline design.** We developed advanced losses and architectures tailored to annotation-free training scenarios: a group-ordering contrastive loss based on differentiable sorting that improves the local neighborhood structure of the embedding space (Chapter 3); a modality-fusion architecture and training objective that enable fusion of any combination of modalities using only noisy multimodal supervision (Chapter 4); local and global objectives and architectures that enable spatio–temporal grounding using only noisy web multimodal supervision, without any grounding annotations (Chapter 9); training-free architectural modifications that enable spatial action grounding in frozen multimodal models (Chapter 10). We also develop pipelines for adapting models to action recognition and text–video retrieval without paired downstream data (Chapter 5 and 6).

11.2 SUMMARY OF KEY CONCLUSIONS

This section revisits the conclusions of each chapter in more detail and outline the connections between chapters, before discussing future work in Section 11.3.

11.2.1 Pretraining Representations Without Human Supervision

In **Part I, Pretraining Representations Without Human Supervision**, we focused on multi-modal and self-supervised learning to develop powerful representations for image and video recognition.

In Chapter 3, we revisited the standard contrastive objective and introduced *group ordering constraints*, implemented via *differentiable sorting* [SPK⁺23]. Our new objective improved the local neighborhood structure of the embedding space and consistently outperformed the pairwise contrastive objective on all k -NN metrics, including retrieval and classification tasks. While our new loss is potentially transferable to multimodal representation learning for pairwise modality alignment, it did not by itself resolve the problem of fusion across multiple modalities.

Therefore, in Chapter 4, we moved beyond pairwise modality alignment in multimodal learning and proposed a *modality-agnostic fusion transformer* that can fuse arbitrary combinations of input modalities, such as video, audio, and text [SCR⁺22]. This transformer was trained with our *combinatorial loss* over all modality subsets, using only noisy web-scale supervision from instructional videos and yielded a strong joint embedding despite noise in text subtitles and imperfect modality alignment, improving retrieval and action localization by leveraging fused modalities. While in this chapter we showed that noisy, naturally co-occurring video–audio–text signals in instructional videos provide a strong basis for learning effective fused representations, further in Chapter 7 (Part III) we showed that automatically improving textual narrations in instructional videos yields significant gains. In Chapter 6 (Part II), we also demonstrated that adapting models to the target domain’s text style, without paired human annotations, further boosts retrieval performance in that domain.

11.2.2 Finetuning Multimodal Models Without Human Supervision

In **Part II, Finetuning Multimodal Models Without Human Supervision**, we focused on finetuning multimodal models for specific tasks and domains without labeled data. Prior work has largely relied on supervised datasets for finetuning. Here we developed the first methods that remove this requirement.

In Chapter 5, we addressed the challenge of adapting image–language models for video understanding, namely zero-shot action recognition and showed that adaptation can be achieved using only unlabeled videos and language priors [LKS⁺23]. Constructing rich *text bags* per video from action dictionaries, large language models world-knowledge, and image captioners, utilizing image–language models for text bag filtering, produced substantial gains in zero-shot and few-shot action recognition and even closed or surpassed the performance of fully supervised baselines. This chapter adapted image–language models to reduce object bias and improve action recognition, however, since action labels are short and well-defined, this approach did not yet address tasks that heavily rely on textual style, e.g, text–video retrieval.

Therefore, in Chapter 6, we studied finetuning vision–language models for text–video retrieval, where models must adapt to diverse text styles across domains. We were the first to address a challenging setting in which only target-domain text queries are available for finetuning and no in-domain videos are provided. We found that the *style* of text queries is a key factor in retrieval and proposed In-Style method that preserves the style of the target text queries and transfers it to uncurated web videos using pretrained vision–language models as pseudo-supervision, leading to substantial performance gains and narrowing the gap to fully supervised training [SKSK23].

11.2.3 Improving Data Quality Without Human Supervision

In **Part III, Improving Data Quality Without Human Supervision**, we investigated strategies to enhance the quality of training and validation data without human input in order to build more robust vision models.

In Chapter 7, we found that improving the textual supervision of large-scale instructional video datasets is crucial for effective multimodal pretraining. We leveraged LLMs to generate high-quality captions from noisy, misaligned ASR narrations and refined temporal alignment using a trained-in-the-loop video–language model, resulting in our new video–audio–text *HowToCaption* dataset, pretraining on which significantly boosted model performance, including modality-agnostic fusion transformer developed in Chapter 4, compared to the original noisy supervision [SKH⁺24].

In Chapter 8, we found that many video benchmarks are limited by representation biases which can unintentionally favor models with good static recognition over activity understanding. We introduced the first scalable method to systematically analyze different representation biases by transforming videos into textual descriptions and analyzing them with LLMs that revealed that most benchmarks are heavily object-centric [SNS⁺25]. We further proposed *UTD-descriptions* and *UTD-splits*, datasets of structured video descriptions and object-debiased test splits for 12 considered video benchmarks, that showed that most current models strongly rely on object cues. We hope that, together, these resources provide a stronger foundation for evaluating and developing models with genuine video understanding capabilities.

11.2.4 Grounding with Multimodal Models Without Human Supervision

In **Part IV, Grounding with Multimodal Models Without Human Supervision**, we investigated strategies to equip multimodal models with action grounding capabilities.

In Chapter 9, we demonstrated that noisy instructional videos can also serve as a resource for learning spatio–temporal action grounding. Specifically, using noisy multimodal data we jointly learned *global* temporally aware representations and *local* fine-grained spatial representations that enabled robust spatio–temporal grounding [CSR⁺24]. We also introduced the *GroundingYouTube* benchmark, which, to our knowledge, is the first to provide dense spatio–temporal point and box annotations for *actions* in untrimmed, multi-action videos.

In Chapter 10, we further revealed that the action grounding capabilities of pretrained multimodal models can be unlocked *without* any finetuning. To our knowledge, our *VideoGEM*, inspired by GEM [BPFK24] was the first training-free method for *action* grounding in videos. Our static–dynamic layer weighting scheme prioritized relevant features for identifying activities that lack a clear physical outline, and the prompt decomposition method reduced object bias present in standard image- and video–language model, which, despite requiring no additional training, surpassed all prior methods that rely on finetuning for action localization.

11.3 FUTURE DIRECTIONS

In this thesis, we focus on advancing vision models without relying on human annotations. Our work is centered on developing methods for learning from annotation-free multimodal and unsupervised data, with the ultimate goal of building robust vision models. In the following, we now outline potential future directions within this scope, discuss broader challenges, and note preliminary progress toward addressing them.

11.3.1 Data: Building Robust Pretraining and Evaluation Corpora

Just as data quality is crucial in supervised training, it is equally important in multimodal and unsupervised training, where noisy, low-quality, or biased samples can hurt generalization and significantly slow down convergence. Therefore, in this section, we outline potential future directions on the data side for developing more effective and robust vision models.

11.3.1.1 *Balancing Diversity, Alignment, and Informativeness*

Background. In Chapter 7, we demonstrated that improving the textual quality of instructional video data leads to substantial performance gains [SKH⁺24]. Recent work has also shown that dataset quality can matter as much as dataset size: LAION-5B introduced CLIP-based filtering and aesthetic scoring [SBV⁺22], which was refined in LAION-Aesthetics. DataComp [GIF⁺23], DINOv2 [ODM⁺23], and DINOv3 [SVS⁺25] all emphasize the importance of constructing balanced, clean, and diverse datasets using techniques such as deduplication, resolution filters, and text filters (e.g., removing overly short, overly long, or uninformative captions), and CLIP-based similarity filtering. Other approaches [LLXH22a, LTH⁺24] aim to improve alignment through re-captioning, where pretrained captioning models generate new, better-aligned textual descriptions.

Future Directions. While aforementioned strategies have demonstrated notable benefits in training efficiency, they also come with drawbacks. For instance, filtering with CLIP scores can discard long-tail examples that include rare but valuable vocabulary. Similarly, removing short or long captions and relying on re-captioning can reduce linguistic diversity. As we found in Chapter 6, the text style of training data is highly important for some downstream tasks, e.g., retrieval [SKSK23], thus, only retaining “clean” detailed captions may not always be beneficial. Moreover, most cleaning and alignment pipelines do not consider *informativeness* of each sample. Our representation-bias analysis pipeline [SNS⁺25] in Chapter 8 could be repurposed for *train-time* curation, for example, to flag or reweight shortcut-prone samples. Finally, many strategies still rely on fixed heuristics rather than leveraging direct model signals about which samples are most informative (as in active learning [EPM⁺24].) A promising direction is to design data curation methods that jointly balance *diversity, alignment, and informativeness*, incorporating *model feedback* rather than relying solely on fixed heuristics.

11.3.1.2 *Longer Video Sequences and Dense Annotations*

Background. Most large-scale video datasets are still dominated by short, clip-centric videos. For example, a popular WebVid [BNVZ21a] dataset contains only short clips with a single high-level description and lacks audio, making it insufficient for learning rich temporal and audio-visual representations. Collecting densely annotated long-form videos with audio from the internet is particularly challenging. Instructional videos, while being a promising source

due to their inherently dense narration, are also very noisy. In Chapter 7 we explored utilizing LLMs to convert subtitles into proper dense captions [SKH⁺24], but this approach is limited to instructional videos only. Recent work has also explored re-captioning strategies for specific clips of long internet videos, such as InternVid [WHL⁺23], VAST [CLW⁺24], and Panda-70M [CSM⁺24], yet these efforts typically cover only sparse segments from longer videos, and relying on image-based captioning models can miss important temporal information. FineVideo [FMT⁺24] takes a step forward by constructing a 44K video dataset using Gemini [CBS⁺25] and GPT [HLG⁺24] models to create detailed descriptions of long videos with multiple quality-control stages, yielding more diverse and balanced supervision that captures not only dense captions but also information about scenes and characters. However, 44K samples remain far too small to serve as a pretraining resource at scale, and building such datasets requires substantial computational resources. Therefore, there is still a need to find scalable ways to curate large-scale data for long-form video understanding.

Future Directions. Building on prior progress in dense annotation of long videos, including our ASR-based method [SKH⁺24] presented in Chapter 7, an important future direction is to investigate effective ways of integrating multimodal signals (e.g., speech, audio cues, contextual text, image captions and short video clip captions) to automatically generate dense supervision for long-form videos at web scale, enabling models to move beyond static shortcuts toward richer temporal understanding. Another promising avenue is to rethink annotation strategies, for example by designing hierarchical approaches that combine sparse high-level supervision (e.g., chapters, summaries) with dense localized signals (e.g., actions, narrations), allowing to keep both global context and fine-grained detail, extending [FMT⁺24].

11.3.1.3 *Toward Genuinely Multimodal Datasets*

Background. While most large-scale multimodal datasets are limited to two modalities, for example, image-text (e.g., LAION [SVB⁺21b, LAI24]), video-text (e.g., WebVid [BNVZ21a]), or video-audio (e.g., AudioSet [GEF⁺17], VGGSound [CXVZ20b]), datasets with more modalities, e.g., video-audio-text datasets, are rare and often constrained. For example, YouTube-based datasets such as HowTo100M [MZA⁺19], HD-VILA [XHZ⁺22], YT-Temporal-180M [ZLH⁺21a], which contain video, audio, and text in the form of ASR subtitles, are dominated by instructional content with speech-heavy audio and noisy text supervision, introducing domain bias. This limitation hinders the development of methods that exploit more than two modalities and highlights the need for large-scale, genuinely multimodal datasets.

Future Directions. Ideally, a multimodal dataset for pretraining vision systems would at minimum integrate visual, auditory, and textual signals in a coherent manner: video frames capture appearance and motion, audio adds cues from ambient sounds, background music, and speech, and text (e.g., dense descriptions, metadata, tags, subtitles) provides high-level and fine-level semantic grounding. Additional modalities, for example, 3D cues such as depth or stereo, may further enrich spatial understanding. But the key challenge is to ensure that all modalities, such as vision, audio, and language, are synchronized and semantically meaningful, for example, by validating and improving textual supervision [SKH⁺24], as demonstrated in Chapter 7. An important future direction is building such datasets at scale, that would enable models to move toward a more holistic multimodal understanding.

11.3.1.4 *Debiasing Evaluation Benchmarks*

Background. In Chapter 8, we addressed *representation bias* in video benchmarks and proposed a first scalable pipeline to detect and mitigate representation biases across three dimensions:

concept bias (e.g., object bias), *temporal* bias (e.g., single-frame bias), and *common-sense vs. dataset* bias [SNS⁺25]. However, our study covered only action recognition and text-to-video retrieval tasks.

Future Directions. Going forward, a potential future direction is to broaden the bias analysis and debiasing presented in Chapter 8 to other downstream tasks, e.g., video question answering, which is nowadays a key evaluation task for multimodal large language models, video captioning, and spatio-temporal grounding. Another direction is to refine *temporal* bias analysis beyond the single-frame vs. uniform multi-frame setting by estimating the minimal temporal context needed and comparing needed architectural mechanisms such as early fusion or late fusion for temporal aggregation. Finally, systematic representation bias analysis can be extended to other modalities, for example, it would be both interesting and important to study undesirable representation biases in image and audio datasets.

11.3.2 Models: Learning Efficient Multimodal Representations

Learning multimodal representations requires not only large amounts of diverse data but also models that can process and integrate this data efficiently. In this section, we further outline future directions for model development for pretraining robust multimodal representations.

11.3.2.1 Objectives for Multimodal Dense Features

Background. In this thesis, we relied on *contrastive objectives* for large-scale multimodal (Chapter 4) and unimodal (Chapter 3) pretraining [SCR⁺22, SPK⁺23]. Contrastive learning is a powerful tool, however, the contrastive loss is usually applied to the image/video/text/audio embeddings, but compressing the image/video/text/audio information into a single embedding limits the model’s ability to learn the fine-grained features needed for dense tasks. *Reconstruction and generative objectives*, e.g. autoencoders [VLBM08, KW13], masked token prediction [HCX⁺22, ZWW⁺21, BDPW21, ADM⁺23], autoregressive next-token prediction [RNSS18, BMR⁺20], or denoising diffusion [HJA20, RBL⁺22, HSG⁺22], train models to predict full or partial content from context. These objectives encourage models to preserve spatial details and capture part-whole structure in the learned features, and typically yield better representations for dense tasks, e.g. detection or semantic segmentation. However, these objectives are still rarely used for *multimodal* pretraining *at web scale*, where they could also improve cross-modal alignment [GFI⁺23, ARG⁺25, GRL⁺22, SMV⁺19]. Moreover, while the success of LLMs is largely based on large-scale autoregressive pretraining, yet many multimodal LLMs simply add non-text inputs on top of pretrained LLMs. They typically feed image/video/audio inputs via adapters/resamplers [WBT⁺24, LZG⁺24, ZLB23b, LLWL23c], leveraging language reasoning but *without* teaching the model to *predict* visual or audio tokens, and therefore miss out on reasoning in these modalities.

Future Directions. One potential future direction is to move beyond the contrastive objective for multimodal pretraining at web scale, e.g., presented in Chapter 4, and to explore generative and reconstruction objectives more deeply in a *multimodal* setup to improve cross-modal alignment and reasoning. For example, autoregressive next-token pretraining could move beyond producing a single-modality output (as in current LLMs or multimodal LLMs that generate *only text* tokens conditioned on other modalities) to generating sequences that *interleave text, vision, and audio tokens*. Such a setup would require the model to predict not only words but also pixels and waveforms, thereby encouraging richer reasoning over video and audio modalities.

11.3.2.2 Reasoning Beyond Single Image/Clip

Background. State-of-the-art multimodal foundation models mostly reason about a *single* image or a *single short* video clip [WBT⁺24, CWW⁺24, WLL⁺24]. Despite the context windows getting bigger, pretraining is still mostly single-image- or clip-centric, and backbones are trained to analyze one item, e.g., an image or a video clip, at a time. As a result, they struggle to compare or combine information across *multiple* images or videos. For example, they often fail to reason over sets of images, decide whether two videos show the same action. This limits applications such as tracking a story across shots or understanding a movie plot that spans many scenes. While there has been some recent progress in adding some multi-image pretraining, for example, in LLaVA-OneVision [LZG⁺24] or VideoLLaMA3 [ZLC⁺25] models, the dominant pretraining data still focuses on single-item reasoning, and robust cross-image or cross-video reasoning remains limited.

Future Directions. In Chapter 7, we used an LLM to reason over many video clips in the text form using their textual narrations, which enabled us to improve our task performance by leveraging broader contextual information [SKH⁺24], looking ahead, we also need models that reason over longer horizons across *multiple* images and videos, not just one image or a short video clip as typically done in recent models [WBT⁺24, CWW⁺24, WLL⁺24]. This points to objectives that reward cross-item comparison and temporal understanding.

11.3.2.3 Efficient Architectures

Background. Vision data, especially video, is highly redundant: neighboring patches in images and consecutive frames in video often contain very similar content [FLH⁺22, RPA⁺21, TSWW22]. Yet most current vision backbones, trained at scale with multimodal and self-supervised methods, still rely on standard ViT architectures that process every patch with heavy self-attention [WLL⁺24, WHZ⁺23, SVS⁺25], which makes training and inference slow and expensive. While multiple efficiency methods have been explored, such as compressing redundant patches into a single token [CZL⁺24], token merging [BFD⁺22], or token pruning [YVA⁺22, RZL⁺21], they are often applied post hoc on trained models and therefore can lead to performance drops or are leveraged mainly for ImageNet-scale pretraining. While there is also a growing interest in applying token reduction methods to multimodal LLMs [ZLC⁺25, LSZ⁺25, XWY⁺25, CGWY25], however, reductions are commonly performed on ViT output tokens after heavy computation has already been spent in the vision encoder. Moreover, efficiency methods are rarely adopted during web-scale pretraining for video or multimodal representation learning, despite their potential to produce more efficient backbones, accelerate training, and lower compute costs.

Future Directions. A promising direction is to integrate efficient image and, in particular, video architectures directly into large-scale multimodal and self-supervised pretraining, such as explored in Chapter 3 and 4. Concretely, encoders could learn to drop, merge, or compress redundant content and allocate compute adaptively. An important research question is how to incorporate these mechanisms into current self-supervised objectives without degrading downstream performance. This could reduce training costs, yield stronger, more scalable multimodal representations, and improve temporal reasoning in videos by utilizing longer videos in pretraining.

11.3.3 Analysis and Adaptation of Learned Representations

Building robust vision models requires understanding the properties and relationships between learned representations and knowing how to transfer them across modalities and tasks. In this section, we discuss future directions in systematic analysis of pretrained representations and annotation-free adaptation to new tasks and domains.

11.3.3.1 *Adaptation of Learned Representations without Annotations*

Background. In Chapter 5 and Chapter 6, we took initial steps toward adapting foundation models to new domains *without* paired in-domain, human-annotated data. For action classification and text-to-video retrieval, we showed that using only unpaired language plus unlabeled videos (Chapter 5) or even only in-domain text queries (Chapter 6) can close a meaningful part of the performance gap to fully supervised finetuning [LKS⁺23, SKSK23]. However, our studies were limited to vision-language modalities and two tasks (classification and retrieval), and methods still assumed access to substantial in-domain data, videos or text.

Future Directions. A natural next step is to extend model adaptation *without* human supervision [LKS⁺23, SKSK23], presented in Chapter 5 and 6, beyond vision-language to additional modalities and beyond classification/retrieval to tasks such as captioning, VQA, or grounding. An interesting future direction is also to probe the *boundaries of the required amount of in-domain data*: quantify how many unlabeled examples (e.g., how much in-domain text) are needed for effective adaptation, namely exploring low-data/few-shot training settings.

11.3.3.2 *Towards Systematic Representation Analysis*

Background. State-of-the-art multimodal and self-supervised methods combine diverse self-supervised objectives with varying multi-stage pretraining recipes, model sizes, architectural choices, dataset sizes and compositions, and data curation heuristics [WLL⁺24, ZGY⁺24, ODM⁺23, SVS⁺25, ZWW⁺21, RKH⁺21d]. Experimental evaluations often mix many of these factors, and reported results are usually obtained under different training characteristics (e.g., training datasets, model sizes, training recipes, etc.) and compute budgets. Moreover, hyperparameter tuning and prompt design are frequently performed on subsets of the downstream datasets themselves, which can inflate scores and undermine the validity of cross-paper comparisons. This makes it difficult to disentangle which factors contribute most to downstream performance and generalization, and which factors are responsible for specific properties of the learned representations.

Future Directions. Therefore, a valuable future direction is to build a controlled comparison suite for self-supervised and multimodal pretraining methods, e.g, presented in Chapter 3 and 4, that varies only one factor at a time (e.g., objective, data, architecture, scale), use the fixed hyperparameter tuning protocol with a limited tuning budget, and pair it with a unified test bench that probes different complementary properties, e.g, dense prediction, temporal reasoning, multimodal alignment, and generalization. Such a comparison suite would enable fairer comparison of different representations and a clearer understanding of their properties, strengths, and weaknesses.

11.3.4 A Broader View on the Topic

Our broader long-term goal is to build robust, efficient vision systems that learn without human labels by using the signals the world already provides (e.g., video, audio, text, depth, and other streams), integrate information from long multimodal inputs, ground their decision with evidence, and adapt to diverse tasks and new, changing environments without additional human supervision. Reaching this goal requires developing learning paradigms for training multimodal foundation models that embed world knowledge and can reason beyond individual instances, comparing and integrating information across multiple multimodal inputs, such as several video and audio tracks, multiple photos or graphics, text data, and other inputs. To operate over such long multimodal contexts, models must be both data- and compute-efficient, they should process redundant information in compact and structured forms, encoding knowledge in ways that retain the richness while reducing unnecessary computation. Importantly, predictions must be grounded in space and time, so that conclusions are interpretable and supported by observable evidence. Ultimately, to build a holistic understanding of the world, such systems must continuously absorb new information, adapt to novel tasks, and remain resilient to distribution shifts – all without relying on manual annotations.

BIBLIOGRAPHY

- [AAA⁺23] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Floren-
cia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat,
et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. Cited on page 1.
- [AABL13] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation
analysis. In *International Conference on Machine Learning*, pages 1247–1255. PMLR, 2013.
Cited on page 16.
- [AAH⁺25] Melika Ayoughi, Samira Abnar, Chen Huang, Chris Sandino, Sayeri Lala, Ee-
shan Gunesh Dhekane, Dan Busbridge, Shuangfei Zhai, Vimal Thilak, Josh Susskind,
et al. How parts assemble into wholes: Learning the relative composition of images.
arXiv preprint arXiv:2506.03682, 2025. Cited on page 14.
- [ABARB20] Elad Amrani, Rami Ben-Ari, Daniel Rotman, and Alex Bronstein. Noise estima-
tion using density estimation for self-supervised multimodal learning. *arXiv preprint*
arXiv:2003.03186, 2020. Cited on pages 44, 46, 50, 52, and 53.
- [ABARB21] Elad Amrani, Rami Ben-Ari, Daniel Rotman, and Alex Bronstein. Noise estimation
using density estimation for self-supervised multimodal learning. *Proceedings of the*
AAAI Conference on Artificial Intelligence, 35(8):6644–6652, 2021. Cited on pages 22
and 94.
- [ABF⁺25] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew
Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2:
Self-supervised video models enable understanding, prediction and planning. *arXiv*
preprint arXiv:2506.09985, 2025. Cited on page 16.
- [ACM⁺22] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes,
Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. Masked siamese
networks for label-efficient learning. In *European Conference on Computer Vision*, pages
456–473. Springer, 2022. Cited on page 14.
- [ADA⁺21] Assaf Arbelle, Sivan Doveh, Amit Alfassy, Joseph Shtok, Guy Lev, Eli Schwartz, Hilde
Kuehne, Hila Barak Levi, Prasanna Sattigeri, Rameswar Panda, et al. Detector-free
weakly supervised grounding by separation. In *Proceedings of the IEEE/CVF International*
Conference on Computer Vision, pages 1801–1812, 2021. Cited on page 128.
- [ADM⁺23] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent,
Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images
with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference*
on Computer Vision and Pattern Recognition, pages 15619–15629, 2023. Cited on pages 15
and 155.
- [AEHKL⁺16] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Bal-
akrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale
video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. Cited on pages
22, 92, and 93.
- [AHWS⁺17] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and
Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the*
IEEE/CVF International Conference on Computer Vision, 2017. Cited on pages 25, 80, 81,
107, and 113.

- [AKB⁺19] Hassan Akbari, Svebor Karaman, Surabhi Bhargava, Brian Chen, Carl Vondrick, and Shih-Fu Chang. Multi-level multimodal common semantic space for image-pharse grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12476–12486, 2019. Cited on pages 8, 122, 130, and 278.
- [AMK⁺20] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *Advances in Neural Information Processing Systems*, volume 33, pages 9758–9770, 2020. Cited on page 46.
- [AMN⁺24] Triantafyllos Afouras, Effrosyni Mavroudi, Tushar Nagarajan, Huiyu Wang, and Lorenzo Torresani. Ht-step: Aligning instructional articles with how-to videos. *Advances in Neural Information Processing Systems*, 36, 2024. Cited on page 93.
- [ARG⁺25] Edson Araujo, Andrew Rouditchenko, Yuan Gong, Saurabhchand Bhati, Samuel Thomas, Brian Kingsbury, Leonid Karlinsky, Rogerio Feris, James R Glass, and Hilde Kuehne. Cav-mae sync: Improving contrastive audio-visual mask autoencoders via fine-grained alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18794–18803, 2025. Cited on pages 18, 19, and 155.
- [ARS⁺20] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 25–37, 2020. Cited on pages 8, 44, 46, 49, 51, 52, 122, 125, and 219.
- [ARV20] Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2020. Cited on pages 14 and 32.
- [AVT16] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. *Advances in Neural Information Processing Systems*, 29, 2016. Cited on page 46.
- [AVT17] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. See, hear, and read: Deep aligned representations. *arXiv preprint arXiv:1706.00932*, 2017. Cited on page 46.
- [AYQ⁺21] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *Advances in Neural Information Processing Systems*, 34:24206–24221, 2021. Cited on pages 4, 8, 17, 19, 32, 44, 46, 48, 49, 83, 122, 125, and 238.
- [AZ17] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 609–617, 2017. Cited on pages 17 and 46.
- [AZ18] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *European Conference on Computer Vision*, pages 435–451, 2018. Cited on page 46.
- [AZ20] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020. Cited on pages 128 and 277.
- [BBY⁺23] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. Cited on pages 291 and 293.

- [BCJ⁺22] Simion-Vlad Bogolin, Ioana Croitoru, Hailin Jin, Yang Liu, and Samuel Albanie. Cross modal retrieval with querybank normalisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5194–5205, 2022. Cited on page 75.
- [BDPW21] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. Cited on pages 4, 15, 18, and 155.
- [BEG⁺22] Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Nieves. Revisiting the "video" in video-language understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2917–2927, 2022. Cited on pages 7, 23, 24, and 106.
- [BEL⁺20] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9922–9931, 2020. Cited on page 16.
- [BFD⁺22] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022. Cited on page 156.
- [BGG14] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101—mining discriminative components with random forests. In *European Conference on Computer Vision*, pages 446–461. Springer, 2014. Cited on page 40.
- [BHA⁺21] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. Cited on pages 75 and 238.
- [BHB19] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in Neural Information Processing Systems*, 32, 2019. Cited on pages 4 and 30.
- [BHX⁺22] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning*, pages 1298–1312. PMLR, 2022. Cited on page 15.
- [BKH16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. Cited on page 47.
- [BL00] Lorraine E Bahrack and Robert Lickliter. Intersensory redundancy guides attentional selectivity and perceptual learning in infancy. *Developmental psychology*, 36(2):190, 2000. Cited on page 1.
- [BMR⁺20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020. Cited on pages 1, 6, 60, 64, 111, 155, 227, and 262.
- [BNVZ21a] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. Cited on pages 4, 7, 17, 21, 22, 44, 46, 48, 52, 69, 75, 81, 83, 153, 154, 230, and 263.
- [BNVZ21b] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021. Cited on pages 93, 98, 99, 101, and 102.

- [BPFK24] Walid Bousselham, Felix Petersen, Vittorio Ferrari, and Hilde Kuehne. Grounding everything: Emerging localization properties in vision-language transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3828–3837, 2024. Cited on pages 9, 20, 26, 135, 136, 137, 138, 139, 143, 144, 145, 146, and 152.
- [BPL21] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. Cited on pages 4, 13, 15, 30, 32, 33, and 38.
- [BRLR22] Ryan Burgert, Kanchana Ranasinghe, Xiang Li, and Michael S Ryoo. Peekaboo: Text to image diffusion models are zero-shot segmentors. *arXiv preprint arXiv:2211.13224*, 2022. Cited on page 16.
- [BSoo] Geoffrey C Bowker and Susan Leigh Star. *Sorting things out: Classification and its consequences*. MIT press, 2000. Cited on pages 1 and 149.
- [BSYS⁺21] Yizhak Ben-Shabat, Xin Yu, Fatemeh Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021. Cited on pages 7 and 106.
- [BTBD20] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast Differentiable Sorting and Ranking. In *International Conference on Machine Learning*, pages 950–959. PMLR, 2020. Cited on page 33.
- [BTS23] Piyush Bagad, Makarand Tapaswi, and Cees GM Snoek. Test of time: Instilling video-language models with a sense of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2503–2516, 2023. Cited on pages 7, 23, and 106.
- [BTZ⁺20] Biagio Brattoli, Joseph Tighe, Fedor Zhdanov, Pietro Perona, and Krzysztof Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4613–4623, 2020. Cited on page 62.
- [BZB22] Adam Botach, Evgenii Zheltonozhskii, and Chaim Baskin. End-to-end referring video object segmentation with multimodal transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4985–4995, 2022. Cited on page 24.
- [C⁺] Ego4D Consortium et al. Egocentric live 4d perception (ego4d) database: A large-scale first-person video database, supporting research in multi-modal machine perception for daily life activity. Cited on page 76.
- [CB23] Tyler A Chang and Benjamin K Bergen. Language model behavior: A comprehensive survey. *arXiv preprint arXiv:2303.11504*, 2023. Cited on page 95.
- [CBJD18] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, pages 132–149, 2018. Cited on pages 14 and 32.
- [CBK20] Junwen Chen, Wentao Bao, and Yu Kong. Activity-driven weakly-supervised spatio-temporal grounding from untrimmed videos. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3789–3797, 2020. Cited on page 26.
- [CBL⁺21] Ioana Croitoru, Simion-Vlad Bogolin, Marius Lordeanu, Hailin Jin, Andrew Zisserman, Samuel Albanie, and Yang Liu. Teachtext: Crossmodal generalized distillation for text-video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11583–11593, 2021. Cited on page 75.

- [CBM]¹⁹ Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2959–2968, 2019. Cited on page 14.
- [CBS]²⁵ Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. Cited on page 154.
- [CCM]¹⁸ Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. Temporally grounding natural sentence in video. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 162–171, 2018. Cited on page 25.
- [CD]¹¹ David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200, 2011. Cited on pages 80, 81, 92, 98, and 99.
- [CDW]²⁴ Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *International Journal of Computer Vision*, 132(1):208–223, 2024. Cited on page 15.
- [CFGH]²⁰ Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. Cited on pages 14, 19, 32, 38, and 39.
- [CGT]²² Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022. Cited on page 20.
- [CGWY]²⁵ Chuanqi Cheng, Jian Guan, Wei Wu, and Rui Yan. Scaling video-language models to 10k frames via hierarchical differential distillation. *arXiv preprint arXiv:2504.02438*, 2025. Cited on page 156.
- [CH_{21a}] Shizhe Chen and Dong Huang. Elaborative rehearsal for zero-shot action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13638–13647, 2021. Cited on pages 62, 66, 67, and 225.
- [CH_{21b}] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. Cited on pages 4, 15, 30, 31, 32, 33, 37, 38, 39, and 214.
- [CH₂₂] Santiago Castro and Fabian Caba Heilbron. Fitclip: Refining large-scale pre-trained image-text models for zero-shot video understanding tasks. *arXiv preprint arXiv:2203.13371*, 2022. Cited on pages 5, 60, and 62.
- [CHEGCN]^{15a} Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Nieves. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015. Cited on pages 107 and 113.
- [CHEGCN]^{15b} Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Nieves. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. Cited on page 124.

- [CHT⁺21] Tsai-Shien Chen, Wei-Chih Hung, Hung-Yu Tseng, Shao-Yi Chien, and Ming-Hsuan Yang. Incremental false negative detection for contrastive learning. *arXiv preprint arXiv:2106.03719*, 2021. Cited on page 14.
- [CJ23] Kevin Clark and Priyank Jaini. Text-to-image diffusion models are zero shot classifiers. *Advances in Neural Information Processing Systems*, 36:58921–58937, 2023. Cited on page 16.
- [CKNH20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020. Cited on pages 4, 13, 14, 19, 20, 30, 31, 32, 33, 37, 38, 39, 211, and 214.
- [CKS⁺20] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems*, 33:22243–22255, 2020. Cited on pages 14, 19, 32, and 40.
- [CLJ21] Pengguang Chen, Shu Liu, and Jiaya Jia. Jigsaw clustering for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. Cited on page 14.
- [CLL⁺23] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhonghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *Large Model Systems Organization*, 2023. Cited on pages 92, 94, 96, 100, and 251.
- [CLTB21] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, pages 1931–1942. PMLR, 2021. Cited on page 95.
- [CLW⁺24] Sihan Chen, Handong Li, Qunbo Wang, Zijia Zhao, Mingzhen Sun, Xinxin Zhu, and Jing Liu. Vast: A vision-audio-subtitle-text omni-modality foundation model and dataset. *Advances in Neural Information Processing Systems*, 36, 2024. Cited on pages 7, 17, 18, 21, 23, 24, 94, 108, 154, 245, and 246.
- [CLW⁺25] Yung-Sung Chuang, Yang Li, Dong Wang, Ching-Feng Yeh, Kehan Lyu, Ramya Raghavendra, James Glass, Lifei Huang, Jason Weston, Luke Zettlemoyer, et al. Meta clip 2: A worldwide scaling recipe. *arXiv preprint arXiv:2507.22062*, 2025. Cited on pages 4, 5, 6, and 21.
- [CLY⁺20] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision*, pages 104–120. Springer, 2020. Cited on pages 17, 18, 19, 46, 48, and 61.
- [CMD⁺21] Jean-Baptiste Cordonnier, Aravindh Mahendran, Alexey Dosovitskiy, Dirk Weissenborn, Jakob Uszkoreit, and Thomas Unterthiner. Differentiable patch selection for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2351–2360, 2021. Cited on page 33.
- [CMK⁺14] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014. Cited on page 40.
- [CMLW19] Zhenfang Chen, Lin Ma, Wenhan Luo, and Kwan-Yee Kenneth Wong. Weakly-supervised spatio-temporally grounding natural sentence in video. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1884–1894,

- Florence, Italy, July 2019. Association for Computational Linguistics. Cited on pages 123 and 124.
- [CMM⁺20] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020. Cited on pages 4, 13, 14, 30, 31, 32, 37, 38, 39, 215, and 276.
- [CMS⁺20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. Cited on page 26.
- [CNB⁺18] João Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arxiv*, 1808.01340, 2018. Cited on pages 20, 107, 113, and 263.
- [CNBH⁺18] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. Cited on pages 65 and 225.
- [CNHZ19] João Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arxiv*, 1907.06987, 2019. Cited on pages 20, 107, 113, and 263.
- [CPR⁺21] Chun-Fu Richard Chen, Rameswar Panda, Kandan Ramakrishnan, Rogerio Feris, John Cohn, Aude Oliva, and Quanfu Fan. Deep analysis of cnn-based spatio-temporal representations for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6165–6175, 2021. Cited on pages 65, 69, and 225.
- [CRC⁺20] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020. Cited on page 15.
- [CRD⁺21] Brian Chen, Andrew Rouditchenko, Kevin Duarte, Hilde Kuehne, Samuel Thomas, Angie Boggust, Rameswar Panda, Brian Kingsbury, Rogerio Feris, David Harwath, et al. Multimodal clustering networks for self-supervised learning from unlabeled videos. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8012–8021, 2021. Cited on pages 8, 25, 44, 46, 49, 50, 51, 52, 53, 94, 100, 102, 122, 125, 217, 218, 220, and 281.
- [CSDS21] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. Cited on pages 22, 83, 99, 101, 102, and 263.
- [CSM⁺24] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13331, 2024. Cited on pages 7, 23, 24, and 154.
- [CSR⁺24] Brian Chen, Nina Shvetsova, Andrew Rouditchenko, Daniel Kondermann, Samuel Thomas, Shih-Fu Chang, Rogerio Feris, James Glass, and Hilde Kuehne. What when and where? self-supervised spatio-temporal grounding in untrimmed multi-action videos from narrated instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18419–18429, 2024. © 2024 IEEE. Reprinted, with permission, from B. Chen et al., "What, When, and Where? Self-Supervised Spatio-

- Temporal Grounding in Untrimmed Multi-Action Videos from Narrated Instructions," in 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2024, pp. 18419–18429, doi: 10.1109/CVPR52733.2024.01743. Cited on pages 3, 8, 9, 11, 26, 43, 121, 136, 137, 138, 143, 144, and 152.
- [CSSL23] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19830–19843, 2023. Cited on page 15.
- [CTM⁺21] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. Cited on pages 4, 13, 14, 15, 19, 30, 32, 37, 38, 39, 76, 211, 212, and 215.
- [CTV19] Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. Differentiable ranking and sorting using optimal transport. *Advances in Neural Information Processing Systems*, 32, 2019. Cited on pages 31 and 33.
- [CTY21] Yi-Wen Chen, Yi-Hsuan Tsai, and Ming-Hsuan Yang. End-to-end multi-modal video temporal grounding. *Advances in Neural Information Processing Systems*, 34:28442–28453, 2021. Cited on page 125.
- [CTZ⁺24] Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, et al. Temporalbench: Benchmarking fine-grained temporal understanding for multimodal video models. *arXiv preprint arXiv:2410.10818*, 2024. Cited on pages 7, 23, and 106.
- [Cut13] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, 2013. Cited on pages 14, 123, 126, and 276.
- [CWL⁺23] Feng Cheng, Xizi Wang, Jie Lei, David Crandall, Mohit Bansal, and Gedas Bertasius. Vindlu: A recipe for effective video-and-language pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10739–10750, 2023. Cited on page 18.
- [CWP⁺20] Ying Cheng, Ruize Wang, Zhihao Pan, Rui Feng, and Yuejie Zhang. Look, listen, and attend: Co-attention network for self-supervised audio-visual representation learning. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3884–3892, 2020. Cited on page 46.
- [CWR22] Jihoon Chung, Yu Wu, and Olga Russakovsky. Enabling detailed action recognition evaluation through video dataset augmentation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 39020–39033. Curran Associates, Inc., 2022. Cited on pages 7, 23, 24, 106, 107, and 111.
- [CWW⁺24] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024. Cited on page 156.
- [CXH21a] Xinlei Chen*, Saining Xie*, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021. Cited on pages 13 and 76.

- [CXH21b] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021. Cited on page 14.
- [CXVZ20a] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 721–725. IEEE, 2020. Cited on page 46.
- [CXVZ20b] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 721–725. IEEE, 2020. Cited on page 154.
- [CZ17] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. Cited on page 50.
- [CZJW20] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10638–10647, 2020. Cited on page 75.
- [CZL⁺24] Rohan Choudhury, Guanglei Zhu, Sihan Liu, Koichiro Niinuma, Kris Kitani, and László Jeni. Don’t look twice: Faster video transformers with run-length tokenization. *Advances in Neural Information Processing Systems*, 37:28127–28149, 2024. Cited on page 156.
- [CZSL20] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. Cited on page 83.
- [DAT⁺21] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021. Cited on pages 4, 14, 19, 30, and 32.
- [DBK⁺20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. Cited on pages 15, 49, and 139.
- [DBK⁺21a] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. Cited on pages 66 and 226.
- [DBK⁺21b] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. Cited on pages 75 and 237.
- [DBZ⁺23] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, Nenghai Yu, and Baining Guo. Peco: Perceptual codebook for bert pre-training of vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 552–560, 2023. Cited on page 15.
- [DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. Cited on pages 54, 55, 94, and 143.

- [DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019. Cited on pages 15, 17, and 18.
- [DCS⁺21] Kevin Duarte, Brian Chen, Nina Shvetsova, Andrew Rouditchenko, Samuel Thomas, Alexander Liu, David Harwath, James Glass, Hilde Kuehne, and Mubarak Shah. Routing with self-attention for multimodal capsule networks. *arXiv preprint arXiv:2112.00775*, 2021. Cited on page 46.
- [DDF⁺22] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, pages 1–23, 2022. Cited on page 124.
- [DDS⁺09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009. Cited on pages 1, 14, and 50.
- [DGRS22] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. Tclr: Temporal contrastive learning for video representation. *Computer Vision and Image Understanding*, 219:103406, 2022. Cited on page 16.
- [DJ21] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11162–11173, 2021. Cited on page 93.
- [DKKP21] Maksim Dzabraev, Maksim Kalashnikov, Stepan Komkov, and Aleksandr Petiushko. Mdmmt: Multidomain multimodal transformer for video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3354–3363, 2021. Cited on page 75.
- [DLH⁺23] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. Mevis: A large-scale benchmark for video segmentation with motion expressions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2694–2703, 2023. Cited on page 24.
- [DLX⁺19] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9346–9355, 2019. Cited on page 17.
- [DSRB14] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in Neural Information Processing Systems*, 27, 2014. Cited on page 14.
- [DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893. Ieee, 2005. Cited on page 15.
- [DXDC13] Pradipto Das, Chenliang Xu, Richard F. Doell, and Jason J. Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2013. Cited on pages 20, 21, 107, and 113.

- [EGH21] Linus Ericsson, Henry Gouk, and Timothy M Hospedales. How well do self-supervised models transfer? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5414–5423, 2021. Cited on page 41.
- [EPM⁺24] Talfan Evans, Shreya Pathak, Hamza Merzic, Jonathan Schwarz, Ryutaro Tanno, and Olivier J Henaff. Bad students make great teachers: Active learning accelerates large-scale visual understanding. In *European Conference on Computer Vision*, pages 264–280. Springer, 2024. Cited on page 153.
- [ESSS21] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International Conference on Machine Learning*, pages 3015–3024. PMLR, 2021. Cited on pages 4, 15, 30, 32, and 38.
- [EVGW⁺10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. Cited on page 41.
- [FA01] József Fiser and Richard N Aslin. Unsupervised statistical learning of higher-order spatial structures from visual scenes. *Psychological science*, 12(6):499–504, 2001. Cited on page 2.
- [FA02] József Fiser and Richard N Aslin. Statistical learning of higher-order temporal structure from visual shape sequences. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 28(3):458, 2002. Cited on page 2.
- [FBGG17] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3636–3645, 2017. Cited on page 16.
- [FCS⁺13] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc’Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in Neural Information Processing Systems*, 26, 2013. Cited on pages 16 and 125.
- [FFFP04] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 178–178. IEEE, 2004. Cited on page 40.
- [FFX⁺21] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3299–3309, 2021. Cited on page 16.
- [FKI⁺23] Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving clip training with language rewrites. *Advances in Neural Information Processing Systems*, 36:35544–35575, 2023. Cited on page 23.
- [FLG⁺21] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*, 2021. Cited on pages 18 and 19.
- [FLH⁺22] Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. Masked autoencoders as spatiotemporal learners. *Advances in Neural Information Processing Systems*, 35:35946–35958, 2022. Cited on pages 16, 20, and 156.
- [FMT⁺24] Miquel Farré, Andi Marafioti, Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. Finevideo. <https://huggingface.co/datasets/HuggingFaceFV/finevideo>, 2024. Cited on page 154.

- [FPNK22] Simone Fabbrizzi, Symeon Papadopoulos, Eirini Ntoutsis, and Ioannis Kompatsiaris. A survey on bias in visual datasets. *Computer Vision and Image Understanding*, 223, 2022. Cited on page 23.
- [FRT⁺21] Andreas Fürst, Elisabeth Rumetshofer, Viet Tran, Hubert Ramsauer, Fei Tang, Johannes Lehner, David Kreil, Michael Kopp, Günter Klambauer, Angela Bitto-Nemling, et al. Cloob: Modern hopfield networks with infoloob outperform clip. *arXiv preprint arXiv:2110.11316*, 2021. Cited on page 62.
- [FXXC21] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*, 2021. Cited on pages 75 and 77.
- [FYH⁺25] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 241–258, Cham, 2025. Springer Nature Switzerland. Cited on page 15.
- [FZF⁺22] Tim Frommknecht, Pedro Alves Zipf, Quanfu Fan, Nina Shvetsova, and Hilde Kuehne. Augmentation learning for semi-supervised classification. In *DAGM German Conference on Pattern Recognition*, pages 85–98. Springer, 2022. Cited on page 12.
- [GAZS⁺25] Gonzalo Martin Garcia, Karim Abou Zeid, Christian Schmidt, Daan De Geus, Alexander Hermans, and Bastian Leibe. Fine-tuning image-conditional diffusion models is easier than you think. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 753–762. IEEE, 2025. Cited on page 15.
- [GBB⁺22] Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan A Rossi, Vishwa Vinay, and Aditya Grover. Cyclop: Cyclic contrastive language-image pretraining. *arXiv preprint arXiv:2205.14459*, 2022. Cited on page 62.
- [GDG19] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019. Cited on page 125.
- [GDSX19] Mingfei Gao, Larry S Davis, Richard Socher, and Caiming Xiong. Wslln: Weakly supervised natural language localization networks. *arXiv preprint arXiv:1909.00239*, 2019. Cited on page 25.
- [GEF⁺17] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 776–780. IEEE, 2017. Cited on pages 83 and 154.
- [GEKM⁺17a] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5842–5850, 2017. Cited on pages 20, 65, and 225.
- [GEKM⁺17b] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thureau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017. Cited on pages 7, 23, 24, 106, 107, 113, and 263.

- [GENL⁺23] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023. Cited on page 17.
- [GFG18] Ruohan Gao, Rogerio Feris, and Kristen Grauman. Learning to separate object sounds by watching unlabeled video. In *European Conference on Computer Vision*, pages 35–53, 2018. Cited on page 17.
- [GFI⁺23] Mariana-Iuliana Georgescu, Eduardo Fonseca, Radu Tudor Ionescu, Mario Lucic, Cordelia Schmid, and Anurag Arnab. Audiovisual masked autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16144–16154, 2023. Cited on pages 18 and 155.
- [GG19] Ruohan Gao and Kristen Grauman. Co-separating sounds of visual objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3879–3888, 2019. Cited on page 17.
- [GGLS18] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees GM Snoek. Actor and action video segmentation from a sentence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5958–5966, 2018. Cited on page 24.
- [GGZ⁺24] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2):581–595, 2024. Cited on page 20.
- [GH10] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010. Cited on pages 127 and 128.
- [GHZ⁺20] Chuang Gan, Deng Huang, Hang Zhao, Joshua B Tenenbaum, and Antonio Torralba. Music gesture for visual sound separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10478–10487, 2020. Cited on page 17.
- [GIF⁺23] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruva Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36:27092–27112, 2023. Cited on pages 7, 17, 21, 22, and 153.
- [GKIL14] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *International Journal of Computer Vision*, 106(2):210–233, 2014. Cited on page 16.
- [GLC⁺21] Zijian Gao, Jingyu Liu, Sheng Chen, Dedan Chang, Hao Zhang, and Jinwei Yuan. Clip2tv: An empirical study on transformer-based methods for video-text retrieval. *arXiv preprint arXiv:2111.05610*, 2021. Cited on page 75.
- [GLKC22] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *International Conference on Learning Representations*, 2022. Cited on pages 5 and 60.
- [GLX⁺22] Yuting Gao, Jinfeng Liu, Zihan Xu, Jun Zhang, Ke Li, and Chunhua Shen. Pyramidclip: Hierarchical feature alignment for vision-language model pretraining. *arXiv preprint arXiv:2204.14095*, 2022. Cited on page 62.

- [GRHD22] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Audioclip: Extending clip to image, text and audio. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 976–980. IEEE, 2022. Cited on page 44.
- [GRL⁺22] Yuan Gong, Andrew Rouditchenko, Alexander H Liu, David Harwath, Leonid Karlinsky, Hilde Kuehne, and James Glass. Contrastive audio-visual masked autoencoder. *arXiv preprint arXiv:2210.07839*, 2022. Cited on pages 18 and 155.
- [GS00] Ken W Grant and Philip-Franz Seitz. The use of visible speech cues for improving auditory detection of spoken sentences. *The Journal of the Acoustical Society of America*, 108(3):1197–1208, 2000. Cited on page 1.
- [GS06] Asif A Ghazanfar and Charles E Schroeder. Is neocortex essentially multisensory? *Trends in cognitive sciences*, 10(6):278–285, 2006. Cited on page 1.
- [GSA⁺20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. Cited on pages 4, 13, 15, 30, 31, 32, 37, and 38.
- [GSAS20] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision*, pages 214–229. Springer, 2020. Cited on pages 44, 47, 52, 75, 81, and 102.
- [GSK18] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. Cited on pages 4, 14, and 31.
- [GSR⁺18] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. Cited on pages 8, 25, 124, 128, 130, 136, 263, 278, and 279.
- [GSYN17] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5267–5275, 2017. Cited on page 25.
- [GTCWP20] Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. Content planning for neural story generation with aristotelian rescoring. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4319–4338, Online, November 2020. Association for Computational Linguistics. Cited on page 250.
- [GTM19] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12046–12055, 2019. Cited on page 93.
- [GVM⁺22] Satya Krishna Gorti, Noël Vouitsis, Junwei Ma, Keyvan Golestan, Maksims Volkovs, Animesh Garg, and Guangwei Yu. X-pool: Cross-modal language-video attention for text-video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5006–5015, 2022. Cited on pages 6 and 75.
- [GWB⁺22] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. Cited on pages 76 and 124.

- [GWH⁺14] Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *European Conference on Computer Vision*, pages 529–545. Springer, 2014. Cited on page 16.
- [GWZE19] Aditya Grover, Eric Wang, Aaron Zweig, and Stefano Ermon. Stochastic Optimization of Sorting Networks via Continuous Relaxations. In *International Conference on Learning Representations*, 2019. Cited on pages 31 and 33.
- [GYMT21] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021. Cited on page 15.
- [HBD⁺20] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. Cited on page 79.
- [HCX⁺22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. Cited on pages 4, 13, 14, 15, 76, and 155.
- [HCZ⁺25] Chunlei Huo, Keming Chen, Shuaihao Zhang, Zeyu Wang, Heyu Yan, Jing Shen, Yuyang Hong, Geqi Qi, Hongmei Fang, and Zihan Wang. When remote sensing meets foundation model: A survey and beyond. *remote sensing*, 17(2), 2025. Cited on page 19.
- [HD24] Shuting He and Henghui Ding. Decoupling static and hierarchical motion perception for referring video segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13332–13341, 2024. Cited on pages 24 and 26.
- [HE20] Dat Huynh and Ehsan Elhamifar. A shared multi-attention framework for multi-label zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8776–8786, 2020. Cited on page 46.
- [HFW⁺20a] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. Cited on pages 4, 19, 20, 30, 31, and 32.
- [HFW⁺20b] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. Cited on pages 13 and 14.
- [HGL⁺25] Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2005–2015, 2025. Cited on page 15.
- [HGX⁺22] Jiaxing Huang, Dayan Guan, Aoran Xiao, Shijian Lu, and Ling Shao. Category contrast for unsupervised domain adaptation in visual tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1203–1214, 2022. Cited on page 15.
- [HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. Cited on pages 15 and 155.

- [HKA⁺21] Olivier J Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron Van den Oord, Oriol Vinyals, and Joao Carreira. Efficient visual pretraining with contrastive detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10086–10096, 2021. Cited on page 14.
- [HKS18] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. Cited on page 50.
- [HKW⁺22] Tri Huynh, Simon Kornblith, Matthew R Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2785–2795, 2022. Cited on pages 14, 32, and 33.
- [HLG⁺24] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. Cited on page 154.
- [HMH⁺25] Alec Helbling, Tuna Han Salih Meral, Ben Hoover, Pinar Yanardag, and Duen Horng Chau. Conceptattention: Diffusion transformers learn highly interpretable features. *arXiv preprint arXiv:2502.04320*, 2025. Cited on page 16.
- [HMLV⁺20] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, et al. spaCy: Industrial-strength Natural Language Processing in Python. *Github*, 2020. Cited on pages 249 and 250.
- [HN21] Lisa Anne Hendricks and Aida Nematzadeh. Probing image-language transformers for verb understanding. *arXiv preprint arXiv:2106.09141*, 2021. Cited on pages 6 and 60.
- [HPL⁺19] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. Cited on page 97.
- [HRS⁺18] David Harwath, Adria Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James Glass. Jointly discovering visual objects and spoken words from raw sensory input. In *European Conference on Computer Vision*, pages 649–665, 2018. Cited on pages 46, 220, and 281.
- [HSG⁺22] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022. Cited on page 155.
- [HSSK16] Yun He, Soma Shirakabe, Yutaka Satoh, and Hirokatsu Kataoka. Human action recognition without human. In *European Conference on Computer Vision*, pages 11–17. Springer, 2016. Cited on pages 7, 23, 24, 106, 107, and 111.
- [HSST04] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004. Cited on page 16.
- [HSW⁺21a] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. Cited on page 20.
- [HSW⁺21b] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. Cited on page 20.

- [HSX⁺23] Po-Yao Huang, Vasu Sharma, Hu Xu, Chaitanya Ryali, Yanghao Li, Shang-Wen Li, Gargi Ghosh, Jitendra Malik, Christoph Feichtenhofer, et al. Mavil: Masked audio-video learners. *Advances in Neural Information Processing Systems*, 36:20371–20393, 2023. Cited on page 18.
- [HWC⁺24] Cheng Han, Qifan Wang, Yiming Cui, Wenguan Wang, Lifu Huang, Siyuan Qi, and Dongfang Liu. Facing the elephant in the room: Visual prompt tuning or full finetuning? *arXiv preprint arXiv:2401.12902*, 2024. Cited on page 20.
- [HXZ19] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019. Cited on page 16.
- [HXZ20] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *European Conference on Computer Vision*, pages 312–329. Springer, 2020. Cited on page 16.
- [HXZ22] Tengda Han, Weidi Xie, and Andrew Zisserman. Temporal alignment networks for long-term video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2906–2916, 2022. Cited on pages 7, 22, 76, 92, 94, 97, 98, 100, 101, 102, 123, and 126.
- [HZC⁺21] Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed Chi. Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. *Advances in Neural Information Processing Systems*, 34, 2021. Cited on page 33.
- [HZL⁺20] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*, 2020. Cited on pages 17 and 18.
- [HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. Cited on pages 37, 50, and 143.
- [IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, 2015. Cited on page 37.
- [ITAC18] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Mining on manifolds: Metric learning without labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. Cited on page 14.
- [IWG⁺22] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. In *Advances in Neural Information Processing Systems*, volume 35, pages 29262–29277, 2022. Cited on page 66.
- [IWW⁺21] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. If you use this software, please cite it as below. Cited on pages 15 and 17.
- [IZJ⁺17] H. Idrees, A. R. Zamir, Y. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah. The thumos challenge on action recognition for videos “in the wild”. *Computer Vision and Image Understanding*, 155:1–23, 2017. Cited on page 124.

- [JBA⁺21] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021. Cited on pages 44 and 49.
- [JGZ⁺13] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3192–3199, 2013. Cited on pages 8, 124, and 136.
- [JHZ⁺22] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *European Conference on Computer Vision*, pages 105–124. Springer, 2022. Cited on pages 5, 20, 21, 60, 62, 66, 67, and 68.
- [JLS⁺24] Wei Ji, Xiangyan Liu, Yingfei Sun, Jiajun Deng, You Qin, Ammar Nuwanna, Mengyao Qiu, Lina Wei, and Roger Zimmermann. Described spatial-temporal video detection. *arXiv preprint arXiv:2407.05610*, 2024. Cited on page 26.
- [JLZ⁺14] Yu-Gang Jiang, Jingen Liu, A Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. Thumos challenge: Action recognition with a large number of classes. <http://crcv.ucf.edu/THUMOS14/>, 2014. Cited on pages 125 and 279.
- [JM24] Yang Jin and Yadong Mu. Weakly-supervised spatio-temporal video grounding with variational cross-modal alignment. In *European Conference on Computer Vision*, pages 412–429. Springer, 2024. Cited on page 26.
- [JSM⁺23] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. Cited on pages 113, 260, and 265.
- [JTC⁺22] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022. Cited on page 20.
- [JylYM22] Yang Jin, yongzhi li, Zehuan Yuan, and Yadong MU. Embracing consistency: A one-stage approach for spatio-temporal video grounding. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. Cited on pages 8, 26, 122, 124, 130, 131, 132, 138, 143, 279, and 280.
- [JYX⁺21a] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, 2021. Cited on pages 17 and 92.
- [JYX⁺21b] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. Cited on pages 60 and 61.
- [JYX⁺21c] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. Cited on page 76.
- [KA24] Yasufumi Kawano and Yoshimitsu Aoki. Maskdiffusion: Exploiting pre-trained diffusion models for semantic segmentation. *IEEE Access*, 2024. Cited on page 15.

- [KAG22] Bulat Khaertdinov, Stylianos Asteriadis, and Esam Ghaleb. Dynamic temperature scaling in contrastive self-supervised learning for sensor-based human activity recognition. *IEEE Transactions On Biometrics, Behavior, And Identity Science*, 2022. Cited on page 14.
- [KARX24] Taha Koleilat, Hojat Asgariandehkordi, Hassan Rivaz, and Yiming Xiao. Medclip-sam: Bridging text and image towards universal medical image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 643–653. Springer, 2024. Cited on page 19.
- [KB15a] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. Cited on pages 51, 83, and 237.
- [KB15b] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. Cited on page 277.
- [KBS⁺23] Anna Kukleva, Moritz Böhle, Bernt Schiele, Hilde Kuehne, and Christian Rupprecht. Temperature schedules for self-supervised contrastive methods on long-tail data. *arXiv preprint arXiv:2303.13664*, 2023. Cited on pages 14 and 19.
- [KCS⁺17] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. Cited on pages 1, 6, 20, 21, 23, 60, 65, 107, 113, 225, and 263.
- [KDSFF13] Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. *Second Workshop on Fine-Grained Visual Categorization*, 2013. Cited on page 40.
- [KGS⁺17] Lukasz Kaiser, Aidan N Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. One model to learn them all. *arXiv preprint arXiv:1706.05137*, 2017. Cited on page 46.
- [KH⁺09] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report*, 2009. Cited on page 40.
- [KHR⁺17] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 706–715, 2017. Cited on page 25.
- [KIRG19] Hilde Kuehne, Ahsan Iqbal, Alexander Richard, and Juergen Gall. Mining youtube-a dataset for learning fine-grained action concepts from webly supervised video data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. Cited on pages 45, 51, 122, 123, 125, 128, 129, 130, 131, 132, 138, 144, 217, 218, and 278.
- [KJG⁺11] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2011. Cited on pages 65, 106, and 225.
- [KJL⁺24] Seongchan Kim, Woojeong Jin, Sangbeom Lim, Heeji Yoon, Hyunwook Choi, and Seungryong Kim. Referring video object segmentation via language-aligned track selection. *arXiv preprint arXiv:2412.01136*, 2024. Cited on pages 24 and 26.
- [KKR25] Akash Kumar, Zsolt Kira, and Yogesh Singh Rawat. Contextual self-paced learning for weakly supervised spatio-temporal video grounding. *arXiv preprint arXiv:2501.17053*, 2025. Cited on page 26.

- [KMSZ12] Alexander Kläser, Marcin Marszałek, Cordelia Schmid, and Andrew Zisserman. Human focused action localization in video. In *Trends and Topics in Computer Vision: ECCV 2010 Workshops, Heraklion, Crete, Greece, September 10–11, 2010, Revised Selected Papers, Part I 11*, pages 219–233. Springer, 2012. Cited on page 136.
- [Knu98] Donald E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching (2nd Ed.)*. Addison Wesley, 1998. Cited on pages 34 and 213.
- [KOH⁺24] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9492–9502, 2024. Cited on page 15.
- [KRJ⁺22] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022. Cited on page 20.
- [KRS18] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *Asian conference on computer vision*, pages 123–141. Springer, 2018. Cited on page 24.
- [KSK21] Wonjae Kim, Bokyoung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021. Cited on pages 17 and 61.
- [KSL⁺21] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021. Cited on page 124.
- [KSP⁺20] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 2020. Cited on page 14.
- [KSSo8] Robyn S Kim, Aaron R Seitz, and Ladan Shams. Benefits of stimulus congruency for multisensory facilitation of visual learning. *PLoS One*, 3(1):e1532, 2008. Cited on page 1.
- [KSZ14] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014. Cited on page 16.
- [KTW⁺20] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020. Cited on page 15.
- [KW13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. Cited on page 155.
- [KW18] Mahnaz Koupaee and William Yang Wang. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*, 2018. Cited on pages 22, 23, 25, 80, 81, 85, 94, and 98.
- [KWFS17] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4405–4413, 2017. Cited on page 129.

- [KYP⁺21] Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9619–9628, 2021. Cited on page 15.
- [KZG⁺17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. Cited on pages 83, 101, and 263.
- [KZPPK19] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. Visil: Fine-grained spatio-temporal video similarity learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6351–6360, 2019. Cited on page 124.
- [LAI24] LAION. Releasing re-laion-5b: transparent iteration on laion-5b with additional safety fixes. <https://laion.ai/blog/re-laion-5b/>, 2024. Accessed: 30 aug, 2024. Cited on pages 7 and 154.
- [LANZ19] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. *arXiv preprint arXiv:1907.13487*, 2019. Cited on page 75.
- [LBB21] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858, 2021. Cited on page 25.
- [LBB23] Jie Lei, Tamara L Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 487–507, Toronto, Canada, July 2023. Association for Computational Linguistics. Cited on pages 7, 23, 24, and 106.
- [LBPL19] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. "vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks". *Advances in Neural Information Processing Systems*, 32, 2019. Cited on pages 17, 18, and 95.
- [LCBH⁺22] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. Cited on page 15.
- [LCC⁺20] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. HERO: Hierarchical encoder for Video+Language omni-representation pre-training. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2046–2065, Online, November 2020. Association for Computational Linguistics. Cited on pages 17, 18, and 76.
- [LCH⁺18] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *European Conference on Computer Vision*, pages 201–216, 2018. Cited on page 127.
- [LCH⁺23] Zeqian Li, Qirui Chen, Tengda Han, Ya Zhang, Yanfeng Wang, and Weidi Xie. A strong baseline for temporal video-text alignment. *arXiv preprint arXiv:2312.14055*, 2023. Cited on pages 23, 24, and 94.
- [LCJ⁺21] Hyunsung Lee, Sangwoo Cho, Yeongjae Jang, Jaekwang Kim, and Honguk Woo. Differentiable ranking metric using relaxed sorting for top-k recommendation. *IEEE Access*, 9:114649–114658, 2021. Cited on page 33.

- [LDF⁺20] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11336–11344, 2020. Cited on pages 17, 18, and 76.
- [LDT⁺21] Shuang Li, Yilun Du, Antonio Torralba, Josef Sivic, and Bryan Russell. Weakly supervised human-object interaction detection in video via contrastive spatiotemporal regions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1845–1855, 2021. Cited on pages 124, 129, 131, 137, 143, and 278.
- [LEP22] Alexander C Li, Alexei A Efros, and Deepak Pathak. Understanding collapse in non-contrastive siamese representation learning. In *European Conference on Computer Vision*, pages 490–505. Springer, 2022. Cited on page 15.
- [LH16] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. Cited on page 37.
- [LH19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. Cited on pages 66, 83, 249, and 264.
- [LHH⁺21] Si Liu, Tianrui Hui, Shaofei Huang, Yunchao Wei, Bo Li, and Guanbin Li. Cross-modal progressive comprehension for referring segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4761–4775, 2021. Cited on page 24.
- [LHH22] Yikang Li, Jenhao Hsiao, and Chiuman Ho. Videoclip: A cross-attention model for fast video-text retrieval task with image clip. In *Proceedings of the 2022 International Conference on Multimedia Retrieval*, pages 29–33, 2022. Cited on pages 19 and 125.
- [LHSY17] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 667–676, 2017. Cited on page 16.
- [LHW⁺23] Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. Cited on page 108.
- [Liu19] Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364, 2019. Cited on page 143.
- [LJS⁺20a] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020. Cited on pages 46 and 53.
- [LJS⁺20b] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020. Cited on pages 18, 25, 76, and 77.
- [LJZ⁺21] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. CLIP4Clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860*, 2021. Cited on pages 44, 47, 52, 217, and 218.
- [LJZ⁺22] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508, 2022. Cited on pages 17, 19, 21, 75, 76, 78, 83, 95, 99, 101, 237, and 238.

- [LKS¹¹] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. Recognizing human actions by attributes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3337–3344. IEEE, 2011. Cited on page 62.
- [LKS⁺₂₃] Wei Lin, Leonid Karlinsky, Nina Shvetsova, Horst Possegger, Mateusz Kozinski, Rameswar Panda, Rogerio Feris, Hilde Kuehne, and Horst Bischof. Match, expand and improve: Unsupervised finetuning for zero-shot action recognition with language knowledge. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2851–2862, 2023. © 2023 IEEE. Reprinted, with permission, from W. Lin et al., "MAtch, eXpand and Improve: Unsupervised Finetuning for Zero-Shot Action Recognition with Language Knowledge," 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2023, pp. 2839–2850, doi: 10.1109/ICCV51070.2023.00267. Cited on pages 3, 6, 10, 20, 21, 59, 151, and 157.
- [LLC⁺₂₁] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. Cited on page 143.
- [LLL⁺₂₂] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. Swinbert: End-to-end transformers with sparse attention for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. Cited on pages 102 and 249.
- [LLL⁺₂₄] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. Cited on pages 111, 113, 259, 260, and 265.
- [LLLL₂₃] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. Cited on page 111.
- [LLLL₂₄] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. Cited on pages 259 and 260.
- [LLSH₂₃] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, 2023. Cited on pages 17, 18, and 95.
- [LLT⁺₂₅] Tianming Liang, Kun-Yu Lin, Chaolei Tan, Jianguo Zhang, Wei-Shi Zheng, and Jian-Fang Hu. Referdino: Referring video object segmentation with visual grounding foundations. *arXiv preprint arXiv:2501.14607*, 2025. Cited on pages 24 and 26.
- [LLV₁₈] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *European Conference on Computer Vision*, 2018. Cited on pages 7, 23, 24, 106, 108, and 110.
- [LLW⁺₂₄] Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. *arXiv preprint arXiv:2403.06977*, 2024. Cited on pages 6, 107, 115, 263, and 264.
- [LLWL_{23a}] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2023. Cited on page 95.
- [LLWL_{23b}] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. Cited on pages 111 and 265.

- [LLWL23c] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023. Cited on page 155.
- [LLXH22a] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022. Cited on pages 4, 6, 7, 18, 19, 21, 22, 23, 60, 61, 65, 153, and 227.
- [LLXH22b] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. Cited on pages 17, 18, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 86, 95, 101, 102, 103, 236, 237, 238, 243, 246, 247, 249, 251, and 288.
- [LLZ⁺21a] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7331–7341, 2021. Cited on pages 48 and 75.
- [LLZ⁺21b] Tianjiao Li, Jun Liu, Wei Zhang, Yun Ni, Wenqian Wang, and Zhiheng Li. Uav-human: A large benchmark for human behavior understanding with unmanned aerial vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16266–16275, 2021. Cited on pages 65 and 226.
- [LLZ⁺21c] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021. Cited on page 17.
- [LLZ⁺21d] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021. Cited on page 62.
- [LMB⁺14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. Cited on pages 16, 62, 83, 101, 125, and 263.
- [LMS16] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European Conference on Computer Vision*, pages 577–593. Springer, 2016. Cited on page 4.
- [LPB⁺22] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13853–13863, 2022. Cited on pages 23, 94, 98, 101, and 102.
- [LPD⁺23] Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2206–2217, 2023. Cited on page 16.
- [LRC⁺23] Vladislav Lialin, Stephen Rawls, David Chan, Shalini Ghosh, Anna Rumshisky, and Wael Hamza. Scalable and accurate self-supervised multimodal representation learning without aligned video and text data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023. Cited on page 95.

- [LSG⁺21a] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in Neural Information Processing Systems*, 34:9694–9705, 2021. Cited on page 17.
- [LSG⁺21b] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in Neural Information Processing Systems*, 34:9694–9705, 2021. Cited on pages 17, 18, and 19.
- [LSZ⁺25] Dongchen Lu, Yuyao Sun, Zilu Zhang, Leping Huang, Jianliang Zeng, Mao Shu, and Huo Cao. Internvl-x: Advancing and accelerating internvl series with efficient visual token compression. *arXiv preprint arXiv:2503.21307*, 2025. Cited on page 156.
- [LSZU16] Renjie Liao, Alex Schwing, Richard Zemel, and Raquel Urtasun. Learning deep parsimonious representations. *Advances in Neural Information Processing Systems*, 29, 2016. Cited on page 14.
- [LTH⁺24] Xianhang Li, Haoqin Tu, Mude Hui, Zeyu Wang, Bingchen Zhao, Junfei Xiao, Sucheng Ren, Jieru Mei, Qing Liu, Huangjie Zheng, et al. What if we recaption billions of web images with llama-3? *arXiv preprint arXiv:2406.08478*, 2024. Cited on pages 23 and 153.
- [LTR⁺20] Ang Li, Meghana Thotakuri, David A Ross, João Carreira, Alexander Vostrikov, and Andrew Zisserman. The ava-kinetics localized human actions video dataset. *arXiv preprint arXiv:2005.00214*, 2020. Cited on page 124.
- [LWB⁺22] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. Cited on page 60.
- [LWDL23] Yi Li, Hualiang Wang, Yiqun Duan, and Xiaomeng Li. Clip surgery for better explainability with enhancement in open-vocabulary tasks. *arXiv preprint arXiv:2304.05653*, 2023. Cited on pages 9, 136, and 138.
- [LWH⁺22] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Yu Qiao. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer. *arXiv preprint arXiv:2211.09552*, 2022. Cited on pages 20 and 263.
- [LWL⁺23] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19948–19960, 2023. Cited on pages 6, 17, 18, 20, 21, 107, 108, 115, 246, 263, and 264.
- [LWN⁺18] Meng Liu, Xiang Wang, Liqiang Nie, Xiangnan He, Baoquan Chen, and Tat-Seng Chua. Attentive moment retrieval in videos. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 15–24, 2018. Cited on page 25.
- [LWZ⁺23] Mengze Li, Han Wang, Wenqiao Zhang, Jiaxu Miao, Zhou Zhao, Shengyu Zhang, Wei Ji, and Fei Wu. Winner: Weakly-supervised hierarchical decomposition and alignment for spatio-temporal video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23090–23099, 2023. Cited on page 26.
- [LXT⁺22] Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, Hehong Chen, Guohai Xu, Zheng Cao, et al. mplug: Effective and efficient vision-language learning by cross-modal skip-connections. *arXiv preprint arXiv:2205.12005*, 2022. Cited on pages 17, 18, 245, and 246.

- [LXX⁺22] Yuqi Liu, Pengfei Xiong, Luhui Xu, Shengming Cao, and Qin Jin. Ts2-net: Token shift and selection transformer for text-video retrieval. In *European Conference on Computer Vision*, pages 319–335. Springer, 2022. Cited on pages 6, 75, and 77.
- [LYBB20] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *European Conference on Computer Vision*, pages 447–463. Springer, 2020. Cited on page 25.
- [LYL⁺20] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020. Cited on page 61.
- [LYY⁺19] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. Cited on pages 17 and 18.
- [LZC⁺23] Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. Univtg: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2794–2804, 2023. Cited on page 25.
- [LZG⁺24] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. Cited on pages 155 and 156.
- [LZR⁺23] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. Cited on page 138.
- [LZZ⁺20] Zhijie Lin, Zhou Zhao, Zhu Zhang, Qi Wang, and Huasheng Liu. Weakly-supervised video moment retrieval via semantic completion network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11539–11546, 2020. Cited on page 25.
- [LZZ⁺22] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. Cited on pages 8, 122, 125, 130, 131, 143, 276, and 279.
- [LZZ⁺24] Zhengfeng Lai, Haotian Zhang, Bowen Zhang, Wentao Wu, Haoping Bai, Aleksei Timofeev, Xianzhi Du, Zhe Gan, Jiulong Shan, Chen-Nee Chuah, et al. Veclip: Improving clip training via visual-enriched captions. In *European Conference on Computer Vision*, pages 111–127. Springer, 2024. Cited on page 23.
- [MAL⁺21] Antoine Miech, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman. Thinking fast and slow: Efficient text-to-visual retrieval with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9826–9836, 2021. Cited on page 238.
- [MAM23] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023. Cited on pages 7, 23, and 106.

- [MAS⁺20] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020. Cited on pages 8, 17, 20, 21, 22, 25, 44, 46, 51, 52, 53, 65, 75, 76, 92, 94, 102, 122, 123, 125, 130, 131, 132, 237, 249, 276, 277, and 279.
- [May05] Richard E Mayer. Cognitive theory of multimedia learning. *The Cambridge handbook of multimedia learning*, 41(1):31–48, 2005. Cited on page 1.
- [MAZ⁺19] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):502–508, 2019. Cited on pages 65, 107, 113, and 226.
- [MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. Cited on pages 50, 130, 237, 276, and 277.
- [MCH20] Jonghwan Mun, Minsu Cho, and Bohyung Han. Local-global video-text interactions for temporal grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10810–10819, 2020. Cited on page 25.
- [MDFB⁺21] Matteo Marucci, Gianluca Di Flumeri, Gianluca Borghini, Nicolina Sciaraffa, Michele Scandola, Enea Francesco Pavone, Fabio Babiloni, Viviana Betti, and Pietro Aricò. The impact of multisensory integration and perceptual load in virtual reality settings on performance, workload and presence. *Scientific Reports*, 11(1):4831, 2021. Cited on page 1.
- [MDRS20] Bruce McIntosh, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. Visual-textual capsule routing for text-based video segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9942–9951, 2020. Cited on page 24.
- [MHP⁺23] WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. Query-dependent video representation for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23023–23033, 2023. Cited on page 25.
- [MJL⁺21] Mathew Monfort, SouYoung Jin, Alexander Liu, David Harwath, Rogerio Feris, James Glass, and Aude Oliva. Spoken moments: Learning joint audio-visual representations from video descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. Cited on pages 107 and 113.
- [MLJ⁺24] Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. Sfrembedding-mistral: enhance text retrieval with transfer learning. *Salesforce AI Research Blog*, 3, 2024. Cited on pages 112, 113, 259, 260, and 262.
- [MLMRC18] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *Proceedings of the 2018 ACM on international conference on multimedia retrieval*, pages 19–27, 2018. Cited on page 17.
- [MLNM19] Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. Generating personalized recipes from historical user preferences. *arXiv preprint arXiv:1909.00105*, 2019. Cited on pages 80, 81, and 85.

- [MLS09] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009. Cited on page 106.
- [MLS17] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. Cited on page 51.
- [MLS18] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv preprint arXiv:1804.02516*, 2018. Cited on page 17.
- [MMRM17] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor’s margins: Local descriptor learning loss. *Advances in Neural Information Processing Systems*, 30, 2017. Cited on pages 15 and 19.
- [MMS19] Shuang Ma, Daniel McDuff, and Yale Song. Unpaired image-to-speech synthesis with multimodal information bottleneck. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7598–7607, 2019. Cited on page 46.
- [MMS⁺21] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021. Cited on page 23.
- [MND⁺19] Devraj Mandal, Sanath Narayan, Sai Kumar Dwivedi, Vikram Gupta, Shuaib Ahmed, Fahad Shahbaz Khan, and Ling Shao. Out-of-distribution detection for generalized zero-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9985–9993, 2019. Cited on page 62.
- [MNXA21] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Open world compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5222–5230, 2021. Cited on page 46.
- [MPPRC18] Niluthpol Chowdhury Mithun, Rameswar Panda, Evangelos E Papalexakis, and Amit K Roy-Chowdhury. Webly supervised joint embedding for cross-modal image-text retrieval. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1856–1864, 2018. Cited on page 17.
- [MPRC19] Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. Weakly supervised video moment retrieval from text queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11592–11601, 2019. Cited on page 25.
- [MRK⁺13] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. Cited on page 40.
- [MRKK23] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023. Cited on pages 23, 24, and 94.
- [MRKK24] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12585–12602, Bangkok, Thailand, August 2024. Association for Computational Linguistics. Cited on page 108.
- [MS17] Pascal Mettes and Cees GM Snoek. Spatial-aware object embeddings for zero-shot localization and classification of actions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4443–4452, 2017. Cited on pages 8 and 136.

- [MTB22] Joanna Materzyńska, Antonio Torralba, and David Bau. Disentangling visual and written concepts in clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16410–16419, 2022. Cited on page 238.
- [Mulo7] *Dynamic Time Warping*, pages 69–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. Cited on page 22.
- [MXS⁺22] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM international conference on multimedia*, pages 638–647, 2022. Cited on page 17.
- [MZA⁺19] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019. Cited on pages 4, 7, 8, 16, 17, 19, 20, 21, 22, 25, 26, 44, 45, 46, 47, 50, 51, 52, 53, 75, 76, 77, 80, 81, 82, 83, 92, 94, 97, 98, 100, 122, 123, 125, 154, 218, 219, 238, 251, 263, 277, and 288.
- [MZH16a] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016. Cited on pages 4 and 16.
- [MZH16b] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Unsupervised learning using sequential verification for action recognition. *arXiv preprint arXiv:1603.08561*, 2(7):8, 2016. Cited on page 16.
- [NF16] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016. Cited on pages 4, 14, and 31.
- [NPC⁺22] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. In *European Conference on Computer Vision*, pages 1–18. Springer, 2022. Cited on pages 5, 20, 60, 62, 66, 67, 68, 225, and 227.
- [NSFK24] Koichi Namekata, Amirmojtaba Sabour, Sanja Fidler, and Seung Wook Kim. Emerdiff: Emerging pixel-level semantic knowledge in diffusion models. *arXiv preprint arXiv:2401.11739*, 2024. Cited on page 16.
- [NSS⁺22] Arsha Nagrani, Paul Hongsuck Seo, Bryan Seybold, Anja Hauth, Santiago Manen, Chen Sun, and Cordelia Schmid. Learning audio-video modalities from image captions. In *European Conference on Computer Vision*, pages 407–426. Springer, 2022. Cited on pages 83, 85, 92, 94, 98, 99, 101, 102, and 103.
- [NXP⁺22] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*, 2022. Cited on page 94.
- [NXWT20] Ke Ning, Lingxi Xie, Fei Wu, and Qi Tian. Polar relative positional encoding for video-language segmentation. In *IJCAI*, volume 9, page 10, 2020. Cited on page 24.
- [NYA⁺21] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. *Advances in Neural Information Processing Systems*, 34:14200–14213, 2021. Cited on page 46.

- [NZo8] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. Cited on page 40.
- [ODM⁺23] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. Cited on pages 13, 14, 16, 22, 153, and 157.
- [OE18] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *European Conference on Computer Vision*, pages 631–648, 2018. Cited on page 17.
- [OKB11] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in Neural Information Processing Systems*, 24, 2011. Cited on pages 83, 101, and 263.
- [OLV18] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. Cited on pages 14, 16, 17, 19, 39, 50, 75, 79, and 95.
- [PBKD21] Felix Petersen, Christian Borgelt, Hilde Kuehne, and Oliver Deussen. Differentiable Sorting Networks for Scalable Sorting and Ranking Supervision. In *International Conference on Machine Learning*, pages 8546–8555. PMLR, 2021. Cited on pages 31, 33, and 34.
- [PBKD22] Felix Petersen, Christian Borgelt, Hilde Kuehne, and Oliver Deussen. Monotonic differentiable sorting networks. *International Conference on Learning Representations*, 2022. Cited on pages 31, 33, 34, and 35.
- [PDB⁺22] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022. Cited on page 15.
- [PE20] Sebastian Prillo and Julian Eisenschlos. Softsort: A continuous relaxation for the argsort operator. In *International Conference on Machine Learning*, pages 7793–7802. PMLR, 2020. Cited on page 33.
- [PHA⁺21] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, Joao Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. In *International Conference on Learning Representations*, 2021. Cited on pages 52, 83, and 238.
- [PHM⁺21] Mandela Patrick, Po-Yao Huang, Ishan Misra, Florian Metze, Andrea Vedaldi, Yuki M Asano, and Joao F Henriques. Space-time crop & attend: Improving cross-modal video representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10560–10572, 2021. Cited on page 16.
- [PKBD22] Felix Petersen, Hilde Kuehne, Christian Borgelt, and Oliver Deussen. Differentiable top-k classification learning. In *International Conference on Machine Learning*, pages 17656–17668. PMLR, 2022. Cited on page 36.
- [PKD⁺16] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. Cited on pages 4, 13, 14, 15, and 31.
- [PMY⁺16] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4594–4602, 2016. Cited on page 16.

- [PQOBTM21] Jesús Andrés Portillo-Quintero, José Carlos Ortiz-Bayliss, and Hugo Terashima-Marín. A straightforward framework for video retrieval using clip. In *Pattern Recognition: 13th Mexican Conference*, pages 3–12. Springer, 2021. Cited on pages 75, 76, 83, and 101.
- [PRB⁺21] Amandalynne Paullada, Inioluwa Deborah Raji, Emily M Bender, Emily Denton, and Alex Hanna. Data and its (dis) contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11), 2021. Cited on page 1.
- [PSY⁺21] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11205–11214, 2021. Cited on page 16.
- [PVZJ12] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3498–3505. IEEE, 2012. Cited on page 40.
- [QLS⁺17] Jie Qin, Li Liu, Ling Shao, Fumin Shen, Bingbing Ni, Jiaxin Chen, and Yunhong Wang. Zero-shot action recognition with error-correcting output codes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2833–2842, 2017. Cited on page 62.
- [QMG⁺21] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6964–6974, 2021. Cited on page 16.
- [QYLH22] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G Hauptmann. Rethinking zero-shot action recognition: Learning from latent atomic actions. In *European Conference on Computer Vision*, pages 104–120. Springer, 2022. Cited on pages 62, 66, and 67.
- [RAY⁺16] Scott Reed, Zeynep Akata, Xincheng Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069. PMLR, 2016. Cited on page 46.
- [Ray23] Partha Pratim Ray. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, 2023. Cited on page 251.
- [RBH⁺21] Andrew Rouditchenko, Angie Boggust, David Harwath, Brian Chen, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Hilde Kuehne, Rameswar Panda, Rogerio Feris, Brian Kingsbury, Michael Picheny, Antonio Torralba, and James Glass. Avlnet: Learning audio-visual language representations from instructional videos. pages 3006–3010, 2021. Cited on pages 4, 15, 17, 19, 44, 46, 47, 49, 50, 51, 52, 53, 94, 102, 125, and 220.
- [RBL⁺22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. Cited on page 155.
- [RCS⁺23] Andrew Rouditchenko, Yung-Sung Chuang, Nina Shvetsova, Samuel Thomas, Rogerio Feris, Brian Kingsbury, Leonid Karlinsky, David Harwath, Hilde Kuehne, and James Glass. C2kd: Cross-lingual cross-modal knowledge distillation for multilingual text-video retrieval. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5. IEEE, 2023. Cited on page 12.
- [RCS]20] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*, 2020. Cited on pages 14, 19, 32, and 211.

- [RDS⁺15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. Cited on pages 37, 38, and 62.
- [RHGS15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015. Cited on page 124.
- [RIT⁺18] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5706–5715, 2018. Cited on pages 38 and 39.
- [RKH⁺21a] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. Cited on pages 2, 4, 5, 6, 15, 16, 19, 20, 21, 22, 60, 61, 63, 66, 67, 68, 69, 226, and 228.
- [RKH⁺21b] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. Cited on pages 6, 17, 25, 32, 74, 75, 76, 77, 78, 82, 86, 126, 130, 131, 137, 150, 217, 237, 238, 259, 260, 276, and 279.
- [RKH⁺21c] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. Cited on pages 46, 47, and 52.
- [RKH⁺21d] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. Cited on pages 92, 95, and 157.
- [RKM⁺22] Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Fine-tuned clip models are efficient video learners. *arXiv preprint arXiv:2212.03640*, 2022. Cited on pages 5, 6, 20, 21, 60, 62, 63, 64, 66, 67, 68, 225, 226, 227, 228, and 232.
- [RKX⁺23] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, 2023. Cited on page 92.
- [RLJ⁺22] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation, August 2022. *arXiv:2208.12242 [cs]*. Cited on page 60.
- [RMK⁺22] Hanoona Rasheed, Muhammad Maaz, Muhammad Uzair Khattak, Salman Khan, and Fahad Shahbaz Khan. Bridging the gap between object and image-level representations for open-vocabulary detection. In *Advances in Neural Information Processing Systems*, volume 35, pages 33781–33794, 2022. Cited on page 60.
- [RMV07] Nikhil Rasiwasia, Pedro J Moreno, and Nuno Vasconcelos. Bridging the gap: Query by semantic example. *IEEE Transactions on Multimedia*, 9(5):923–938, 2007. Cited on page 16.

- [RNSS18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. Cited on pages 26 and 155.
- [RPA⁺21] Michael S Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: What can 8 learned tokens do for images and videos? *arXiv preprint arXiv:2106.11297*, 2021. Cited on page 156.
- [RRS15] Anna Rohrbach, Marcus Rohrbach, and Bernt Schiele. The long-short story of movie description. In *German conference on pattern recognition*, pages 209–221. Springer, 2015. Cited on pages 80, 81, 92, 98, and 99.
- [RRTS15] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015. Cited on pages 20, 21, 107, and 113.
- [RRW⁺13] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzl, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1:25–36, 2013. Cited on page 25.
- [RSM⁺22] Yangjun Ruan, Saurabh Singh, Warren Morningstar, Alexander A Alemi, Sergey Ioffe, Ian Fischer, and Joshua V Dillon. Weighted ensemble self-supervised learning. *arXiv preprint arXiv:2211.09981*, 2022. Cited on page 14.
- [RSR⁺20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020. Cited on page 94.
- [RWC⁺19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019. Cited on pages 15, 75, 76, and 94.
- [RZC⁺22] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18082–18091, 2022. Cited on pages 5 and 60.
- [RZL⁺21] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in Neural Information Processing Systems*, 34:13937–13949, 2021. Cited on page 156.
- [SAN96] Jenny R Saffran, Richard N Aslin, and Elissa L Newport. Statistical learning by 8-month-old infants. *science*, 274(5294):1926–1928, 1996. Cited on page 2.
- [SBF⁺21] Nina Shvetsova, Bart Bakker, Irina Fedulova, Heinrich Schulz, and Dmitry V Dylov. Anomaly detection in medical imaging with deep perceptual autoencoders. *IEEE Access*, 9:118571–118583, 2021. Cited on page 12.
- [SBT⁺25] Nina Shvetsova, Goutam Bhat, Prune Truong, Hilde Kuehne, and Federico Tombari. M2svid: End-to-end inpainting and refinement for monocular-to-stereo video conversion. *arXiv preprint arXiv:2505.16565*, 2025. Cited on page 12.
- [SBV⁺22] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. Cited on pages 16, 17, 22, 137, and 153.

- [SCR⁺22] Nina Shvetsova, Brian Chen, Andrew Rouditchenko, Samuel Thomas, Brian Kingsbury, Rogerio S Feris, David Harwath, James Glass, and Hilde Kuehne. Everything at once—multi-modal fusion transformer for video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20020–20029, 2022. © 2022 IEEE. Reprinted, with permission, from N. Shvetsova et al., "Everything at Once – Multi-modal Fusion Transformer for Video Retrieval," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 19988–19997, doi: 10.1109/CVPR52688.2022.01939. Cited on pages 2, 5, 8, 10, 15, 19, 43, 74, 75, 76, 79, 81, 82, 83, 84, 92, 94, 102, 103, 122, 125, 151, 155, 237, 238, 249, and 281.
- [SDGS18] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. Cited on pages 22, 52, 85, and 263.
- [SFW⁺23] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. Cited on page 17.
- [SHG⁺22] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650, 2022. Cited on pages 17, 18, and 19.
- [SHLM22] Bowen Shi, Wei-Ning Hsu, Kushal Lakhota, and Abdelrahman Mohamed. Learning audio-visual speech representation by masked multimodal cluster prediction. *arXiv preprint arXiv:2201.02184*, 2022. Cited on page 18.
- [SIS15] Khurram Soomro, Haroon Idrees, and Mubarak Shah. Action localization in videos through context walk. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3280–3288, 2015. Cited on page 136.
- [SIS18] Khurram Soomro, Haroon Idrees, and Mubarak Shah. Online localization and prediction of actions and interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):459–472, 2018. Cited on page 136.
- [SJL13] Ling Shao, Simon Jones, and Xuelong Li. Efficient search and localization of human actions in video databases. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(3):504–512, 2013. Cited on page 136.
- [SKH⁺24] Nina Shvetsova, Anna Kukleva, Xudong Hong, Christian Rupprecht, Bernt Schiele, and Hilde Kuehne. Howtocation: Prompting llms to transform video annotations at scale. In *European Conference on Computer Vision*. Springer, 2024. First published in Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds) *Computer Vision – ECCV 2024*. ECCV 2024. Lecture Notes in Computer Science, vol 15114. Springer, Cham. https://doi.org/10.1007/978-3-031-72992-8_1. Reproduced with permission from Springer Nature. Cited on pages 3, 8, 11, 24, 43, 91, 108, 152, 153, 154, 156, and 288.
- [SKP15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015. Cited on pages 32 and 210.
- [SKSK23] Nina Shvetsova, Anna Kukleva, Bernt Schiele, and Hilde Kuehne. In-style: Bridging text and uncurated videos with style transfer for text-video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. © 2023 IEEE. Reprinted, with permission, from N. Shvetsova, A. Kukleva, B. Schiele and H. Kuehne, "In-Style: Bridging Text and Uncurated Videos with Style Transfer for Text-Video Retrieval," 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2023,

- pp. 21924–21935, doi: 10.1109/ICCV51070.2023.02009. Cited on pages 3, 6, 11, 20, 21, 43, 73, 108, 151, 153, and 157.
- [SLC⁺18] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1134–1141. IEEE, 2018. Cited on page 16.
- [SLH20] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *European Conference on Computer Vision*, pages 208–223. Springer, 2020. Cited on page 24.
- [SLP⁺21] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. Cited on pages 219 and 220.
- [SLS⁺20] Jonathan C Stroud, Zhichao Lu, Chen Sun, Jia Deng, Rahul Sukthankar, Cordelia Schmid, and David A Ross. Learning video representations from textual web supervision. *arXiv preprint arXiv:2007.14937*, 2020. Cited on pages 22, 92, and 93.
- [SME20] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. Cited on page 15.
- [SMV⁺19] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019. Cited on pages 95, 108, and 155.
- [SNAS22] Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. End-to-end generative pretraining for multimodal video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. Cited on pages 102 and 249.
- [SNS⁺25] Nina Shvetsova, Arsha Nagrani, Bernt Schiele, Hilde Kuehne, and Christian Rupprecht. Unbiasing through textual descriptions: Mitigating representation bias in video benchmarks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 29050–29059, 2025. © 2025 IEEE. Reprinted, with permission, from N. Shvetsova, A. Nagrani, B. Schiele, H. Kuehne and C. Rupprecht, "Unbiasing through Textual Descriptions: Mitigating Representation Bias in Video Benchmarks," 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2025, pp. 29050–29059, doi: 10.1109/CVPR52734.2025.02705. Cited on pages 3, 8, 11, 92, 106, 152, 153, and 155.
- [SP54] William H Sumby and Irwin Pollack. Visual contribution to speech intelligibility in noise. *The journal of the acoustical society of america*, 26(2):212–215, 1954. Cited on page 1.
- [SPA⁺21] Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba Heilbron, Chen Zhao, Silvio Giancola, and Bernard Ghanem. Mad: A scalable dataset for language grounding in videos from movie audio descriptions. *arXiv preprint arXiv:2112.00431*, 2021. Cited on page 125.
- [SPK⁺23] Nina Shvetsova, Felix Petersen, Anna Kukleva, Bernt Schiele, and Hilde Kuehne. Learning by sorting: Self-supervised learning with group ordering constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16453–16463, 2023. © 2023 IEEE. Reprinted, with permission, from N. Shvetsova, F. Petersen, A. Kukleva, B. Schiele and H. Kuehne, "Learning by Sorting: Self-supervised Learning with Group Ordering Constraints," 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2023, pp. 16407–16417, doi: 10.1109/ICCV51070.2023.01508. Cited on pages 2, 5, 10, 19, 29, 151, and 155.

- [SQL20] Hao Shao, Shengju Qian, and Yu Liu. Temporal interlacing network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11966–11973, 2020. Cited on page 62.
- [SRAH⁺17] Rakshith Shetty, Marcus Rohrbach, Lisa Anne Hendricks, Mario Fritz, and Bernt Schiele. Speaking the same language: Matching machine to human captions by adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4135–4144, 2017. Cited on page 250.
- [SRS⁺23] Sirnam Swetha, Mamshad Nayeem Rizve, Nina Shvetsova, Hilde Kuehne, and Mubarak Shah. Preserving modality structure improves multi-modal learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21993–22003, 2023. Cited on pages 12 and 43.
- [SSo8] Ladan Shams and Aaron R Seitz. Benefits of multisensory learning. *Trends in cognitive sciences*, 12(11):411–417, 2008. Cited on page 1.
- [SSCC22] Anshul Shah, Suvrit Sra, Rama Chellappa, and Anoop Cherian. Max-margin contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8220–8230, 2022. Cited on page 38.
- [SVB⁺21a] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. Cited on pages 7, 82, 83, 92, 101, and 102.
- [SVB⁺21b] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. Cited on page 154.
- [SVS⁺25] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. Cited on pages 14, 22, 153, 156, and 157.
- [SVW⁺16] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016. Cited on pages 65 and 226.
- [SWY⁺24] Quan Sun, Jinsheng Wang, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, and Xinlong Wang. Eva-clip-18b: Scaling clip to 18 billion parameters. *arXiv preprint arXiv:2402.04252*, 2024. Cited on pages 4, 6, 17, and 21.
- [SXGX19] Jing Shi, Jia Xu, Boqing Gong, and Chenliang Xu. Not all frames are equal: Weakly-supervised video grounding with contextual similarity and visual clustering losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10444–10452, 2019. Cited on pages 24, 26, 122, 125, and 136.
- [SYX21] Rui Su, Qian Yu, and Dong Xu. Stvgbert: A visual-linguistic transformer based framework for spatio-temporal video grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1533–1542, 2021. Cited on pages 26 and 124.
- [SZC⁺20] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2020. Cited on pages 76 and 95.

- [SZS12] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. Cited on pages 25, 65, 124, 136, and 225.
- [SZS17] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arxiv*, 1705.06950, 2017. Cited on pages 106, 107, and 113.
- [TB19] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. Cited on pages 17, 18, 61, and 95.
- [TDR⁺19] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019. Cited on pages 8, 122, and 124.
- [TE11] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011. Cited on pages 1, 23, and 149.
- [TGZ⁺23] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models*, 2023. Cited on page 94.
- [TJB⁺22] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, 2022. Cited on pages 6, 60, and 62.
- [TJW⁺23] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems*, 36:1363–1389, 2023. Cited on page 16.
- [TKI20] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European Conference on Computer Vision*, pages 776–794. Springer, 2020. Cited on page 14.
- [TKM20] Antigoni Tsiami, Petros Koutras, and Petros Maragos. Stavis: Spatio-temporal audiovisual saliency network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4766–4776, 2020. Cited on page 46.
- [TL21] Mamatha Thota and Georgios Leontidis. Contrastive domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2209–2218, 2021. Cited on pages 15 and 19.
- [TLB21] Zineng Tang, Jie Lei, and Mohit Bansal. Decembert: Learning from noisy instructional videos via dense captions and entropy minimization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2415–2426, 2021. Cited on page 48.
- [TLI⁺23] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. Cited on page 95.
- [TLL⁺21] Zongheng Tang, Yue Liao, Si Liu, Guanbin Li, Xiaojie Jin, Hongxu Jiang, Qian Yu, and Dong Xu. Human-centric spatio-temporal video grounding with visual transformers. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021. Cited on pages 25, 26, and 124.

- [TLWB21] Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. Vimpac: Video pre-training via masked token prediction and contrastive learning. *arXiv preprint arXiv:2106.11250*, 2021. Cited on page 16.
- [TPS⁺21] Reuben Tan, Bryan Plummer, Kate Saenko, Hailin Jin, and Bryan Russell. Look at what i'm doing: Self-supervised spatial grounding of narrations in instructional videos. *Advances in Neural Information Processing Systems*, 34, 2021. Cited on pages 8, 25, 26, 122, 123, 124, 125, 128, 129, 130, 131, 132, 136, 137, 138, 143, 144, 276, 277, and 279.
- [TSF⁺16] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. Cited on page 22.
- [TSP⁺20] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020. Cited on pages 4 and 30.
- [TSSW22] Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17918–17928, 2022. Cited on page 75.
- [TSWW22] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in Neural Information Processing Systems*, 35:10078–10093, 2022. Cited on pages 16, 19, 20, 107, 115, 156, 263, and 264.
- [TWL⁺21] Mingkan Tang, Zhanyu Wang, Zhenhua Liu, Fengyun Rao, Dian Li, and Xiu Li. Clip4caption: Clip for video caption. In *Proceedings of the 29th ACM international conference on multimedia*, pages 4858–4862, 2021. Cited on pages 99, 102, and 249.
- [TXSP21] Reuben Tan, Huijuan Xu, Kate Saenko, and Bryan A Plummer. Logan: Latent graph co-attention network for weakly-supervised video moment retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2083–2092, 2021. Cited on page 25.
- [TZC⁺10] Tuan Hue Thi, Jian Zhang, Li Cheng, Li Wang, and Shinichi Satoh. Human action recognition and localization in video using structured learning of local space-time features. In *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 204–211. IEEE, 2010. Cited on page 136.
- [TZY23] Jiajin Tang, Ge Zheng, and Sibe Yang. Temporal collection and distribution for referring video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15466–15476, 2023. Cited on pages 24 and 26.
- [VBK⁺25] Felix Vogel, Walid Bousselham, Anna Kukleva, Nina Shvetsova, and Hilde Kuehne. Videogem: Training-free action grounding in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3374–3383, 2025. © 2025 IEEE. Reprinted, with permission, from F. Vogel, W. Bousselham, A. Kukleva, N. Shvetsova and H. Kuehne, "VideoGEM: Training-Free Action Grounding in Videos," 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2025, pp. 3374–3383, doi: 10.1109/CVPR52734.2025.00320. Cited on pages 4, 9, 12, 26, and 135.
- [VCS⁺16] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016. Cited on page 79.

- [VLBMo8] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*, pages 1096–1103, 2008. Cited on pages 4, 15, and 155.
- [VLL⁺10] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010. Cited on pages 4 and 15.
- [VSKK22] Felix Vogel, Nina Shvetsova, Leonid Karlinsky, and Hilde Kuehne. VI-taboo: An analysis of attribute-based zero-shot capabilities of vision-language models. *arXiv preprint arXiv:2209.06103*, 2022. Cited on pages 137 and 146.
- [VSP⁺17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. Cited on pages 47, 48, 127, and 232.
- [WBD⁺23] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19175–19186, 2023. Cited on pages 15 and 18.
- [WBT⁺24] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. Cited on pages 155 and 156.
- [WBTT22] Jue Wang, Gedas Bertasius, Du Tran, and Lorenzo Torresani. Long-short temporal contrastive learning of video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14010–14020, 2022. Cited on page 16.
- [WBU11] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011. Cited on pages 16 and 125.
- [WC17] Qian Wang and Ke Chen. Alternative semantic representations for zero-shot human action recognition. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 87–102. Springer, 2017. Cited on page 62.
- [WCL⁺23] Junke Wang, Dongdong Chen, Chong Luo, Xiyang Dai, Lu Yuan, Zuxuan Wu, and Yu-Gang Jiang. Chatvideo: A tracklet-centric multimodal and versatile video understanding system. *arXiv preprint arXiv:2304.14407*, 2023. Cited on page 108.
- [WDMY20] Hao Wang, Cheng Deng, Fan Ma, and Yi Yang. Context modulated dynamic networks for actor and action video segmentation with language queries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12152–12159, 2020. Cited on page 24.
- [WDYT19] Hao Wang, Cheng Deng, Junchi Yan, and Dacheng Tao. Asymmetric cross-guided attention network for actor and action video segmentation from natural language query. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3939–3948, 2019. Cited on page 24.
- [WFW⁺23] Haochen Wang, Junsong Fan, Yuxi Wang, Kaiyou Song, Tong Wang, and ZHAO-XIANG ZHANG. Droppos: Pre-training vision transformers by reconstructing dropped positions. *Advances in Neural Information Processing Systems*, 36:46134–46151, 2023. Cited on page 14.

- [WFX⁺22] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678, 2022. Cited on page 15.
- [WGX21] Wei Wang, Junyu Gao, and Changsheng Xu. Weakly-supervised video object grounding via stable context learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 760–768, 2021. Cited on pages 24 and 26.
- [WGY⁺23] Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Kevin Qinghong Lin, Satoshi Tsutsui, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, et al. All in one: Exploring unified video-language pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6598–6608, 2023. Cited on pages 18, 107, 115, and 263.
- [WHL⁺23] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *International Conference on Learning Representations*, 2023. Cited on pages 6, 7, 17, 20, 21, 22, 23, 24, 94, 107, 108, 115, 137, 154, and 263.
- [WHS15] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3164–3172, 2015. Cited on page 136.
- [WHZ⁺23] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14549–14560, 2023. Cited on pages 16, 20, 107, 115, 156, 263, and 264.
- [WIK⁺22] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022. Cited on page 66.
- [WJS⁺22] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4974–4984, 2022. Cited on page 24.
- [WLF⁺23] Wenhao Wu, Haipeng Luo, Bo Fang, Jingdong Wang, and Wanli Ouyang. Cap4video: What can auxiliary captions do for text-video retrieval? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10704–10713, 2023. Cited on pages 6, 75, and 237.
- [WLHL18] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):394–407, 2018. Cited on page 16.
- [WLL16] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5005–5013, 2016. Cited on page 16.
- [WLL⁺22] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11686–11695, 2022. Cited on pages 8 and 122.

- [WLL⁺24] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, pages 396–416. Springer, 2024. Cited on pages 4, 6, 16, 18, 20, 156, and 157.
- [WLX⁺22] Zhenhailong Wang, Manling Li, Ruochen Xu, Luowei Zhou, Jie Lei, Xudong Lin, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Derek Hoiem, et al. Language models with image descriptors are strong few-shot video-language learners. *Advances in Neural Information Processing Systems*, 35:8483–8497, 2022. Cited on page 108.
- [WLY⁺25] Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025. Cited on page 25.
- [WLZF18] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8052–8060, 2018. Cited on page 16.
- [WMS16] Philippe Weinzaepfel, Xavier Martin, and Cordelia Schmid. Human action localization with sparse spatial supervision. *arXiv preprint arXiv:1605.05197*, 2016. Cited on pages 25, 26, 124, 129, 136, 137, and 144.
- [WNK⁺24] Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Videogrounding-dino: Towards open-vocabulary spatio-temporal video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18909–18918, 2024. Cited on pages 26, 136, 138, and 143.
- [WR21] Philippe Weinzaepfel and Grégory Rogez. Mimetics: Towards understanding human actions out of context. *International Journal of Computer Vision*, 129, 2021. Cited on pages 7, 23, and 106.
- [WSO23] Wenhao Wu, Zhun Sun, and Wanli Ouyang. Revisiting classifier: Transferring vision-language models for video recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7–8, 2023. Cited on pages 5, 60, 62, and 66.
- [WWC⁺19] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. VateX: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4581–4591, 2019. Cited on pages 80 and 81.
- [WWC⁺23] Yuting Wang, Jinpeng Wang, Bin Chen, Ziyun Zeng, and Shu-Tao Xia. Contrastive masked autoencoders for self-supervised video hashing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2733–2741, 2023. Cited on page 16.
- [WWSY20] Chen Wei, Huiyu Wang, Wei Shen, and Alan Yuille. Co2: Consistent contrast for unsupervised visual representation learning. *arXiv preprint arXiv:2010.02217*, 2020. Cited on page 14.
- [WWW⁺21] Guangrun Wang, Keze Wang, Guangcong Wang, Philip HS Torr, and Liang Lin. Solving inefficiency of self-supervised representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. Cited on page 14.
- [WWZ⁺23a] Zhanyu Wang, Longyue Wang, Zhen Zhao, Minghao Wu, Chenyang Lyu, Huayang Li, Deng Cai, Luping Zhou, Shuming Shi, and Zhaopeng Tu. Gpt4video: A unified multimodal large language model for instruction-followed understanding and safety-aware generation. *arXiv preprint arXiv:2311.16511*, 2023. Cited on page 108.

- [WWZ⁺23b] Dongming Wu, Tiancai Wang, Yuang Zhang, Xiangyu Zhang, and Jianbing Shen. Onlinerefer: A simple online baseline for referring video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2761–2770, 2023. Cited on pages 24 and 26.
- [WXL21] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. Cited on pages 5, 6, 20, 21, 60, 62, 66, 67, and 68.
- [WXW⁺16] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, 2016. Cited on page 109.
- [WYH⁺22] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022. Cited on page 102.
- [WYH⁺23] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*, 2023. Cited on pages 112, 259, and 260.
- [WZS⁺21] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021. Cited on page 14.
- [WZY21a] Xiaohan Wang, Linchao Zhu, and Yi Yang. T2vlad: Global-local sequence alignment for text-video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5079–5088, 2021. Cited on page 47.
- [WZY21b] Xiaohan Wang, Linchao Zhu, and Yi Yang. T2vlad: global-local sequence alignment for text-video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5079–5088, 2021. Cited on page 75.
- [XGF16] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, pages 478–487. PMLR, 2016. Cited on page 14.
- [XGH⁺21] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6787–6800, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. Cited on pages 17, 25, 76, and 137.
- [XHE⁺10] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3485–3492. IEEE, 2010. Cited on page 41.
- [XHZ⁺22] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5036–5045, 2022. Cited on pages 22, 92, 94, and 154.
- [XLG⁺21] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2021. Cited on pages 46 and 49.

- [XMYR16] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. Cited on pages 20, 21, 45, 51, 80, 81, 92, 93, 98, 99, 107, 113, and 237.
- [XSH⁺18] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *European Conference on Computer Vision*, pages 305–321, 2018. Cited on pages 52, 82, 130, 276, and 277.
- [XSLP20] Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. Hard negative examples are hard, but useful. In *European Conference on Computer Vision*, pages 126–142. Springer, 2020. Cited on page 210.
- [XWED20] Tete Xiao, Xiaolong Wang, Alexei A Efros, and Trevor Darrell. What should not be contrastive in contrastive learning. *arXiv preprint arXiv:2008.05659*, 2020. Cited on page 32.
- [XWY⁺25] Ling Xing, Alex Jinpeng Wang, Rui Yan, Xiangbo Shu, and Jinhui Tang. Vision-centric token compression in large language model. *arXiv preprint arXiv:2502.00791*, 2025. Cited on page 156.
- [XXCC15] Ran Xu, Caiming Xiong, Wei Chen, and Jason Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015. Cited on page 16.
- [XXT⁺23] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. *arXiv preprint arXiv:2309.16671*, 2023. Cited on pages 5, 7, and 22.
- [XXZ⁺19] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019. Cited on page 16.
- [XYY⁺23] Haiyang Xu, Qinghao Ye, Ming Yan, Yaya Shi, Jiabo Ye, Yuanhong Xu, Chenliang Li, Bin Bi, Qi Qian, Wei Wang, et al. mplug-2: A modularized multi-modal foundation model across text, image and video. *arXiv preprint arXiv:2302.00402*, 2023. Cited on pages 17, 18, 19, 21, 102, and 246.
- [XZC⁺22] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022. Cited on page 15.
- [YBK⁺23a] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *International Conference on Learning Representations*, 2023. Cited on pages 6, 60, and 62.
- [YBK⁺23b] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *International Conference on Learning Representations*, 2023. Cited on pages 137 and 146.
- [YBZ⁺22] Xufeng Yao, Yang Bai, Xinyun Zhang, Yuechen Zhang, Qi Sun, Ran Chen, Ruiyu Li, and Bei Yu. Pcl: Proxy-based contrastive learning for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7097–7107, 2022. Cited on page 15.

- [YDT⁺22] Jinyu Yang, Jiali Duan, Son Tran, Yi Xu, Sampath Chanda, Liquan Chen, Belinda Zeng, Trishul Chilimbi, and Junzhou Huang. Vision-language pre-training with triple contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15671–15680, 2022. Cited on page 61.
- [YGG17] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. Cited on pages 37 and 214.
- [YGW⁺22] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. Unitab: Unifying text and box outputs for grounded vision-language modeling. In *European Conference on Computer Vision*, pages 521–539. Springer, 2022. Cited on page 122.
- [YGZ⁺23] Liangzhe Yuan, Nitesh Bharadwaj Gundavarapu, Long Zhao, Hao Zhou, Yin Cui, Lu Jiang, Xuan Yang, Menglin Jia, Tobias Weyand, Luke Friedman, et al. Videoglue: Video general understanding evaluation of foundation models. *arXiv preprint arXiv:2307.03166*, 2023. Cited on pages 116 and 264.
- [YHH⁺21] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021. Cited on pages 17 and 62.
- [YHH⁺22] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. In *European Conference on Computer Vision*, 2022. Cited on page 14.
- [YHW⁺22] Lewei Yao, Jianhua Han, Youpeng Wen, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, Chunjing Xu, and Hang Xu. Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. *Advances in Neural Information Processing Systems*, 35:9125–9138, 2022. Cited on page 125.
- [YKK18] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *European Conference on Computer Vision*, pages 471–487, 2018. Cited on pages 51 and 75.
- [YLH⁺25] Ziang Yan, Zhilin Li, Yinan He, Chenting Wang, Kunchang Li, Xinhao Li, Xiangyu Zeng, Zilei Wang, Yali Wang, Yu Qiao, et al. Task preference optimization: Improving multimodal large language models with vision task alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 29880–29892, 2025. Cited on page 25.
- [YMH⁺22] Yuncong Yang, Jiawei Ma, Shiyuan Huang, Long Chen, Xudong Lin, Guangxing Han, and Shih-Fu Chang. Tempclr: Temporal alignment representation with contrastive learning. *arXiv preprint arXiv:2212.13738*, 2022. Cited on page 25.
- [YMS⁺22] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Tubedetr: Spatio-temporal video grounding with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16442–16453, 2022. Cited on pages 8, 26, 122, 124, 130, 131, 132, 138, 143, 279, and 280.
- [YNL⁺24] Antoine Yang, Arsha Nagrani, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vidchapters-7m: Video chapters at scale. *Advances in Neural Information Processing Systems*, 36, 2024. Cited on pages 22 and 94.
- [YNS⁺23] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. Cited on page 102.

- [YPB16] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2016. Cited on page 14.
- [YVA⁺22] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10809–10818, 2022. Cited on page 156.
- [YW23] Xingyi Yang and Xinchao Wang. Diffusion model as representation learner. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18938–18949, 2023. Cited on page 16.
- [YWV⁺22] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. Cited on page 76.
- [YXY⁺23] Qinghao Ye, Guohai Xu, Ming Yan, Haiyang Xu, Qi Qian, Ji Zhang, and Fei Huang. Hitea: Hierarchical temporal-aware video-language pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15405–15416, 2023. Cited on page 102.
- [YZW⁺22] Shen Yan, Tao Zhu, Zirui Wang, Yuan Cao, Mi Zhang, Soham Ghosh, Yonghui Wu, and Jiahui Yu. Video-text modeling with zero-shot transfer from contrastive captioners. *arXiv preprint arXiv:2212.04979*, 2022. Cited on pages 101 and 102.
- [YZYC19] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6210–6219, 2019. Cited on pages 4 and 30.
- [ZAC⁺19a] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. Cited on pages 8, 122, 124, and 125.
- [ZAC⁺19b] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3537–3545, 2019. Cited on pages 25, 26, 45, 51, 53, 218, and 219.
- [ZAI⁺22] Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, et al. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*, 2022. Cited on page 108.
- [ZC17] Rowan Zellers and Yejin Choi. Zero-shot activity recognition with verb attribute induction. *arXiv preprint arXiv:1707.09468*, 2017. Cited on page 62.
- [ZCK⁺23] Abhay Zala, Jaemin Cho, Satwik Kottur, Xilun Chen, Barlas Oguz, Yashar Mehdad, and Mohit Bansal. Hierarchical video-moment retrieval and step-captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. Cited on pages 93 and 244.
- [ZCS⁺23] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. Cited on pages 100 and 243.

- [ZDW⁺19] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1247–1257, 2019. Cited on page 25.
- [ZFI⁺22] David Zimmerer, Peter M Full, Fabian Isensee, Paul Jäger, Tim Adler, Jens Petersen, Gregor Köhler, Tobias Ross, Annika Reinke, Antanas Kascenas, et al. Mood 2020: A public benchmark for out-of-distribution detection and localization on medical images. *IEEE transactions on medical imaging*, 41(10):2728–2738, 2022. Cited on page 12.
- [ZFZ⁺22] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. In *European Conference on Computer Vision*, 2022. Cited on pages 5 and 60.
- [ZGJ⁺22] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *European Conference on Computer Vision*, pages 350–368. Springer, 2022. Cited on page 60.
- [ZGMT19] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. The sound of motions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1735–1744, 2019. Cited on page 17.
- [ZGR⁺18] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *European Conference on Computer Vision*, pages 570–586, 2018. Cited on page 17.
- [ZGY⁺24] Long Zhao, Nitesh B Gundavarapu, Liangzhe Yuan, Hao Zhou, Shen Yan, Jennifer J Sun, Luke Friedman, Rui Qian, Tobias Weyand, Yue Zhao, et al. Videoprism: A foundational visual encoder for video understanding. *arXiv preprint arXiv:2402.13217*, 2024. Cited on pages 108 and 157.
- [ZHH⁺23] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *Advances in Neural Information Processing Systems*, 36:45533–45547, 2023. Cited on page 16.
- [ZHZ⁺25] Pengcheng Zhao, Zhixian He, Fuwei Zhang, Shujin Lin, and Fan Zhou. Ld-detr: Loop decoder detection transformer for video moment retrieval and highlight detection. *arXiv preprint arXiv:2501.10787*, 2025. Cited on page 25.
- [ZIE16] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016. Cited on pages 4, 13, 14, and 31.
- [ZJM⁺21] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. Cited on pages 4, 13, 15, 30, 32, 33, and 38.
- [ZJM⁺22] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text. In *Machine learning for healthcare conference*, pages 2–25. PMLR, 2022. Cited on page 17.
- [ZKC⁺19] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. Grounded video description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. Cited on page 124.

- [ZLB23a] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. Cited on page 108.
- [ZLB23b] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. Cited on page 155.
- [ZLC18a] Luowei Zhou, Nathan Louis, and Jason J Corso. Weakly-supervised video object grounding from text by loss weighting and object interaction. *arXiv preprint arXiv:1805.02834*, 2018. Cited on pages 24 and 26.
- [ZLC18b] Luowei Zhou, Nathan Louis, and Jason J Corso. Weakly-supervised video object grounding from text by loss weighting and object interaction. In *British Machine Vision Conference*, 2018. Cited on pages 123 and 125.
- [ZLC⁺25] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025. Cited on page 156.
- [ZLD22] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, pages 696–712. Springer, 2022. Cited on pages 9, 20, 136, and 138.
- [ZLH⁺21a] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *Advances in Neural Information Processing Systems*, 34:23634–23651, 2021. Cited on pages 4, 18, 22, 76, 92, 94, 154, and 263.
- [ZLH⁺21b] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588, 2021. Cited on page 46.
- [ZLI⁺24] Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple LLM framework for long-range video question-answering. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21715–21737, Miami, Florida, USA, November 2024. Association for Computational Linguistics. Cited on page 2.
- [ZLL⁺23] Tianyi Zhang, Shangqing Lyu, Yanli Lei, Sicheng Chen, Nan Ying, Yufang He, Yu Zhao, Yunlu Feng, Hwee Kuan Lee, and Guanglei Zhang. Puzzletuning: Explicitly bridge pathological and natural image with puzzles. *arXiv preprint arXiv:2311.06712*, 2023. Cited on page 14.
- [ZLW⁺23] Yuanhao Zhai, Ziyi Liu, Zhenyu Wu, Yi Wu, Chunluan Zhou, David Doermann, Junsong Yuan, and Gang Hua. Soar: Scene-debiasing open-set action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10244–10254, 2023. Cited on page 23.
- [ZMKB23] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. Cited on pages 4, 5, and 17.
- [ZMKG22] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *arXiv preprint arXiv:2212.04501*, 2022. Cited on pages 6, 75, and 76.

- [ZMKG23] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. Cited on page 95.
- [ZPFL20] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12870–12877, 2020. Cited on page 25.
- [ZSJZ20] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. *arXiv preprint arXiv:2004.13931*, 2020. Cited on page 25.
- [ZSJZ23] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Temporal sentence grounding in videos: A survey and future directions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10443–10465, 2023. Cited on page 25.
- [ZTTY19] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. Hacs: Human action clips and segments dataset for recognition and temporal localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8668–8678, 2019. Cited on page 124.
- [ZWF⁺18] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara L Berg. Visual to sound: Generating natural sound for videos in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3550–3558, 2018. Cited on page 46.
- [ZWM⁺22] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022. Cited on page 17.
- [ZWW⁺21] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. Cited on pages 4, 14, 15, 155, and 157.
- [ZXC18] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. Cited on pages 45, 51, 80, 81, 92, 93, 98, 99, 124, 125, 129, 138, and 144.
- [ZXH⁺20] Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. Dense regression network for video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10287–10296, 2020. Cited on page 125.
- [ZXL⁺20] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6688–6697, 2020. Cited on page 14.
- [ZXQ⁺24] Ximiao Zhang, Min Xu, Dehui Qiu, Ruixin Yan, Ning Lang, and Xiuzhuang Zhou. Mediclip: Adapting clip for few-shot medical image anomaly detection. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 458–468. Springer, 2024. Cited on page 19.
- [ZY20] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8746–8755, 2020. Cited on pages 25, 52, and 53.

- [ZYLL22a] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. Cited on pages 5, 20, and 60.
- [ZYLL22b] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. Cited on pages 20 and 60.
- [ZYW⁺21] Fangneng Zhan, Yingchen Yu, Rongliang Wu, Kaiwen Cui, Aoran Xiao, Shijian Lu, and Ling Shao. Bi-level feature alignment for versatile image translation and manipulation. *arXiv preprint arXiv:2107.03021*, 2021. Cited on page 33.
- [ZYW⁺24] Heng Zhao, Zhao Yinjie, Bihan Wen, Yew-Soon Ong, and Joey Zhou. Video-text prompting for weakly supervised spatio-temporal video grounding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19494–19505, 2024. Cited on page 26.
- [ZZY⁺22] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luwei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16793–16803, 2022. Cited on pages 8, 122, 125, 130, 131, 132, 143, 276, and 279.
- [ZZD⁺25] Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-clip: Unlocking the long-text capability of clip. In *European Conference on Computer Vision*, pages 310–325. Springer, 2025. Cited on pages 259 and 260.
- [ZZH⁺22] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *Advances in Neural Information Processing Systems*, 35:36067–36080, 2022. Cited on page 125.
- [ZZL21a] Yan Zeng, Xinsong Zhang, and Hang Li. Multi-grained vision language pre-training: Aligning texts with visual concepts. *arXiv preprint arXiv:2111.08276*, 2021. Cited on page 18.
- [ZZL⁺21b] Rui Zhu, Bingchen Zhao, Jingen Liu, Zhenglong Sun, and Chang Wen Chen. Improving contrastive learning by visualizing feature transformation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10306–10315, 2021. Cited on page 14.
- [ZZL⁺23] Yan Zeng, Xinsong Zhang, Hang Li, Jiawei Wang, Jipeng Zhang, and Wangchunshu Zhou. X2-vlm: All-in-one pre-trained model for vision-language tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3156–3168, 2023. Cited on page 18.
- [ZZP⁺22] Chaoning Zhang, Kang Zhang, Trung X Pham, Axi Niu, Zhinan Qiao, Chang D Yoo, and In So Kweon. Dual temperature helps contrastive learning without many negative samples: Towards understanding and simplifying moco. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. Cited on page 14.
- [ZZW⁺24] Kecheng Zheng, Yifei Zhang, Wei Wu, Fan Lu, Shuailei Ma, Xin Jin, Wei Chen, and Yujun Shen. Dreamlip: Language-image pre-training with long captions. In *European Conference on Computer Vision*, pages 73–90. Springer, 2024. Cited on page 23.
- [ZZZ⁺20] Zhu Zhang, Zhou Zhao, Yang Zhao, Qi Wang, Huasheng Liu, and Lianli Gao. Where does it exist: Spatio-temporal video grounding for multi-form sentences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10668–10677, 2020. Cited on pages 25, 26, 123, 124, and 128.

- [ZZZ⁺22] Tiancheng Zhao, Tianqi Zhang, Mingwei Zhu, Haozhan Shen, Kyusong Lee, Xiaopeng Lu, and Jianwei Yin. VI-checklist: Evaluating pre-trained vision-language models with objects, attributes and relations. *arXiv preprint arXiv:2207.00221*, 2022. Cited on pages 6, 60, and 62.
- [ZZZL21] Yang Zhao, Zhou Zhao, Zhu Zhang, and Zhijie Lin. Cascaded prediction network via segment tree for temporal video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4197–4206, 2021. Cited on page 125.

APPENDIX – LEARNING BY SORTING: SELF-SUPERVISED LEARNING WITH GROUP ORDERING CONSTRAINTS

This appendix contains supplementary material related to Chapter 3, *Learning by Sorting: Self-supervised Learning with Group Ordering Constraints*, of this thesis. We first discuss relations between the GroCo loss, the contrastive loss, and the triplet loss in Section A.1. Then we provide additional experimental evaluation results in Section A.2 and qualitative analysis in Section A.3. Then we describe odd-even sorting networks in Section A.4. Finally, we cover additional implementation details in Section A.5.

Contents

A.1	Discussion of GroCo/Contrastive/Triplet Loss Relations	209
A.2	Additional Experimental Results	211
A.3	Qualitative Analysis of Learned Representation Space	212
A.4	Odd-even Sorting Network	213
A.5	Implementation Details	214

A.1 DISCUSSION OF GROCO/CONTRASTIVE/TRIPLET LOSS RELATIONS

In this section, we discuss the similarities and differences between the GroCo loss, the contrastive loss, and the triplet loss. For comparison purposes, let's consider a simplified version of losses when there is only one positive example x^p and one negative example x^n for the anchor x^a . We denote the distance from the anchor x^a to the positive sample x^p as $d^p = -\frac{x^a \top x^p}{\|x^a\| \|x^p\|}$ and the distance from the anchor x^a to the negative sample x^n as $d^n = -\frac{x^a \top x^n}{\|x^a\| \|x^n\|}$. Then contrastive InfoNCE loss (with respect to the anchor x^a) is defined as:

$$\begin{aligned} L_{\text{Contrastive}} &= -\log \frac{\exp(-d^p/\tau)}{\exp(-d^p/\tau) + \exp(-d^n/\tau)} \\ &= \log(1 + \exp(-(d^n - d^p)/\tau)), \end{aligned} \quad (\text{A.1})$$

where τ is a temperature hyperparameter (Figure A.1a).

The triplet loss is defined as:

$$\begin{aligned} L_{\text{Triplet}} &= \max(d^p - d^n + r, 0) = \\ &= \max(r - (d^n - d^p), 0), \end{aligned} \quad (\text{A.2})$$

where r is a margin hyperparameter (Figure A.1b).

For the GroCo loss, a permutation matrix $P \in R^{2 \times 2}$ corresponds to only one conditional

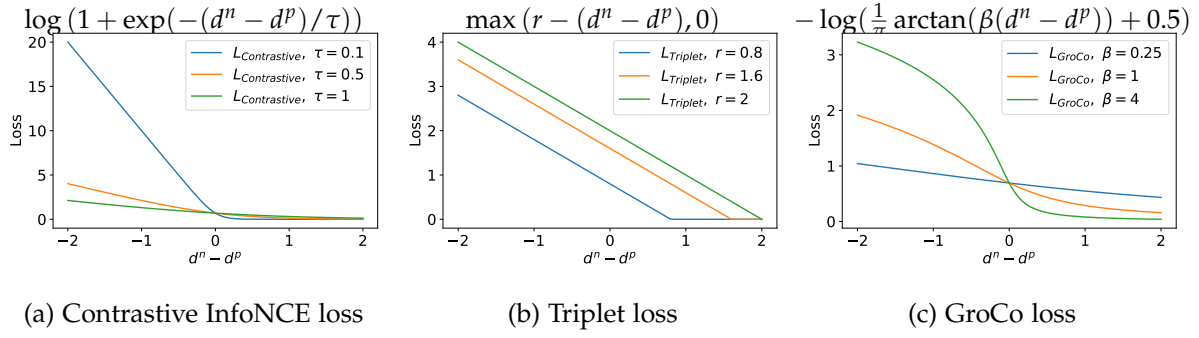


Figure A.1: Comparison of the contrastive loss, the triplet loss, and the GroCo loss in a simple scenario with only one positive example and one negative example for an anchor image. We denote the distance from the anchor to the positive sample as d^p and the distance from the anchor to the negative sample as d^n . We note that in the simple case of only one positive and one negative, all three losses try to maximize the difference between the distances to the positive and negative examples ($d^n - d^p$). The temperature τ , the margin r , or the inverse temperature β define the flatness of the loss curve depending on the difference ($d^n - d^p$).

swap operation and is defined as:

$$\begin{aligned} P_{11} = P_{22} &= f(d^n - d^p) = \frac{1}{\pi} \arctan(\beta(d^n - d^p)) + 0.5, \\ P_{12} = P_{21} &= f(d^p - d^n) = \frac{1}{\pi} \arctan(\beta(d^p - d^n)) + 0.5, \end{aligned} \quad (\text{A.3})$$

where β is an inverse temperature. Therefore, the GroCo loss is defined as:

$$\begin{aligned} L_{GroCo} &= \frac{1}{4} \left(-2 \log \left(\frac{1}{\pi} \arctan(\beta(d^n - d^p)) + 0.5 \right) - \right. \\ &\quad \left. -2 \log \left(1 - \frac{1}{\pi} \arctan(\beta(d^p - d^n)) - 0.5 \right) \right) = \\ &= -\log \left(\frac{1}{\pi} \arctan(\beta(d^n - d^p)) + 0.5 \right), \end{aligned} \quad (\text{A.4})$$

where β is an inverse temperature hyperparameter (Figure A.1c).

In Figure A.1, we show the loss curves with different values of respective hyperparameters. We note that in this simplified example with only one positive and only one negative, all three losses try to maximize the difference between the distances to the positive and negative examples ($d^n - d^p$). The temperature τ , the margin r , or the inverse temperature β define the flatness of the loss curve depending on the difference ($d^n - d^p$).

However, in the case with more negative/positive examples for the anchor image, different losses integrate information from multiple negatives/positives in different ways. For triplet loss, there are various strategies to sample one positive example and one negative example for the anchor image [SKP15, XSLP20]. The complete loss is defined as the sum (or average) of the losses for the chosen triplets $\sum_{ij} \max(r - (d_i^n - d_j^p), 0)$. On the other hand, the contrastive loss aggregates multiple negatives by contrasting the positive example to all negative examples, resulting in sum under logarithm: $\log(1 + \sum_i \exp(-(d_i^n - d^p)/\tau))$. And in contrast to an explicit sum over a predefined number of negatives, the GroCo loss aggregates multiple

positives and negatives via the permutation matrix, conditionally swapping neighboring elements, and later applies the group ordering supervision, enforcing the distance between positive and negative groups.

A.2 ADDITIONAL EXPERIMENTAL RESULTS

In this section, we provide additional experimental evaluations:

Augmentation Strategy. In Table A.1a we evaluate the performance of the model with respect to different augmentation strategies for view sampling. We follow two setups: 1) the augmentation strategy as used in the SimCLR [CKNH20] method with a random resized crop, color jittering, and gaussian blur, grayscaling and horizontal flip and 2) the augmentation strategy as used in the DINO [CTM⁺21] method that extends the SimCLR list of augmentations with solarization. SimCLR augmentations are considered as “stronger” compared to DINO augmentations since they include a larger range of cropping sizes (8% -100% of original image compared to 14%-100% in DINO augmentations) and larger range values in color jittering. We observe that the stronger SimCLR [CKNH20] augmentations are more beneficial for the SimCLR method than the weaker DINO augmentations, while for the proposed method, the DINO augmentation strategy is more beneficial. However, the difference between augmentation strategies diminishes with increasing number of training epochs and is no longer measurable at 400 epochs. For a fair comparison, we use the SimCLR augmentation strategy in all reproductions of the SimCLR method reported in the main part of the thesis.

Projection Dimensionality. We also ablate our method with respect to the dimensionality of the projection space (or the latent space), where distances between samples are computed to calculate a training loss. Table A.1b shows that increasing dimensionality of the projection space increases performance in general, which is more noticeable for the k -NN performance. Note that we do not change the dimensionality of the embedding space (output space of the encoder that is used for the k -NN evaluation and linear evaluation), which is always 2048-dimensional.

Importance of Negatives. We also evaluate the importance of utilizing strong negatives for the successful training of our model. We train the model using ten random negatives instead of the top-10 strongest negatives as a negative group and report performance in Table A.1c. We observe that leveraging the strongest negatives increases performance across all metrics, demonstrating the importance of hard negatives during training with the GroCo loss, similarly as the contrastive loss benefits from hard negative sampling [RCSJ20].

k -NN in Projection Space. We also evaluate the k -NN performance in the projection space (or the latent space) where the training loss is applied. We compare k -NN performance in the projection and representation spaces in Table A.1d. We observe that for both methods, k -NN performance is higher if we use embeddings from the representation space even though we train the model to compare embeddings in the projection space. This could be explained by the fact that the embedding space contains more general image representations since the representations in projection space could be overfitted to the respective augmentations and there become agnostic to some image attributes (like color, since we train the model to match views with different color jittering parameters).

Method	Augmentations	Epochs	k -NN Evaluation			Linear Evaluation	
			$k=1$	$k=10$	$k=20$	Top-1	Top-5
SimCLR	as in SimCLR [CTM ⁺ 21]	100	46.0	51.5	51.9	65.7	86.7
SimCLR	as in DINO [CTM ⁺ 21]	100	43.3	48.6	49.1	63.7	85.4
GroCo	as in SimCLR [CTM ⁺ 21]	100	54.0	59.0	59.4	68.4	88.3
GroCo	as in DINO [CTM ⁺ 21]	100	55.3	60.3	60.5	69.2	88.4
GroCo	as in SimCLR [CTM ⁺ 21]	200	56.7	61.6	61.8	69.8	89.1
GroCo	as in DINO [CTM ⁺ 21]	200	57.7	62.4	62.7	70.4	89.5
GroCo	as in SimCLR [CTM ⁺ 21]	400	58.3	63.3	63.8	71.1	89.7
GroCo	as in DINO [CTM ⁺ 21]	400	58.7	63.4	63.6	71.0	89.7

(a) Augmentation strategy

Projection dim	Embedding dim	k -NN Evaluation			Linear Probing	
		$k=1$	$k=10$	$k=20$	Top-1	Top-5
128	2048	53.7	58.5	58.7	68.1	88.0
512	2048	55.2	59.8	60.1	69.0	88.5
2048	2048	55.3	60.3	60.5	69.2	88.4

(b) Projection dimentionality

	k -NN Evaluation			Linear Probing	
	$k=1$	$k=10$	$k=20$	Top-1	Top-5
10 random negatives	39.5	45.0	45.3	60.1	82.7
top-10 strongest negatives	55.3	60.3	60.5	69.2	88.4

(c) Importance of negatives

Method	Space	k -NN Evaluation		
		$k=1$	$k=10$	$k=20$
SimCLR	Projection Space	35.8	41.6	42.3
SimCLR	Representation Space	46.0	51.5	51.9
GroCo	Projection Space	51.4	56.9	57.3
GroCo	Representation Space	55.3	60.3	60.5

(d) k -NN evaluationTable A.1: **Additional Experiments.** The best results are bolded. Options used to obtain the main results are highlighted. **Backbone=Resnet50, Views=2×224, #epochs=100.**

A.3 QUALITATIVE ANALYSIS OF LEARNED REPRESENTATION SPACE

We also additionally perform a qualitative analysis of learned representation. In Figure A.2, we visualize representations for images from four classes of different types of cats and four classes of different types of dogs. We find that our method produces much more visually separable clusters with respect to “inter-class” variations (cats vs dogs) and “intra-class” variations (between different classes of cats) than the SimCLR baseline.

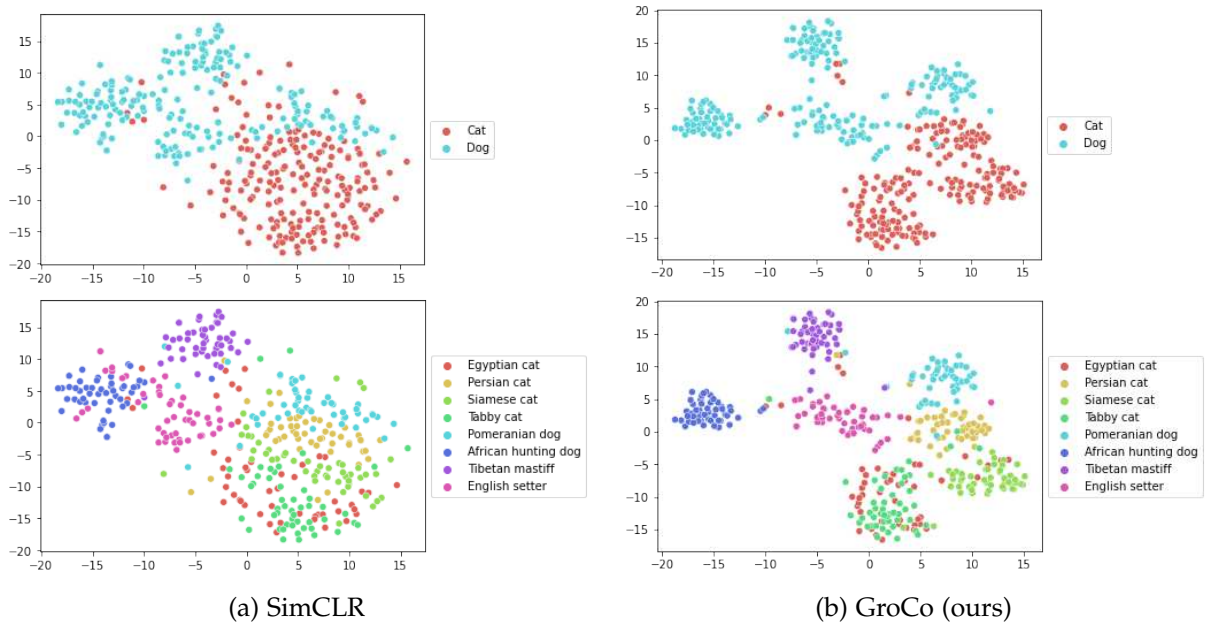


Figure A.2: t-SNE visualization of learned representations of Imagenet validation images from four classes of different types of cats (Egyptian cat, Persian cat, Siamese cat, Tabby cat) and four classes different types of dogs (Pomeranian dog, African hunting dog, Tibetan mastiff, English setter) for the SimCLR method and the proposed method. For visualization we use models with Resnet50 encoder trained for 100 epochs with a batch size of 1024 and 2×224 views.

A.4 ODD-EVEN SORTING NETWORK

An odd-even sorting network, or odd-even sort, is a sorting algorithm from classic computer science literature [Knu98]. Sorting networks, or networks for sorting, are a family of sorting algorithms that consist of the *fixed* sequence of comparisons, in a sense that the next comparisons (elements on which positions are compared) does not depend on the result of previous comparisons. An odd-even sorting network is a simple example of this family of algorithms. The odd-even sorting network compares neighbored elements starting from odd and even indices alternating on each step, and requires n steps to sort a sequence of n elements. We present a pseudocode of the odd-even sorting network in Algorithm 1. We also additionally illustrate the odd-even sorting process in Figure A.3.

Algorithm 1 Python pseudocode of an odd-even sorting network for sorting an array of numbers in non-descending order

```
# arr: array to sort
# n: length of array

for s in range(1, n + 1):
    if s % 2 == 1:
        for i in range(0, n - 1, 2):
            if arr[i] > arr[i+1]:
                arr[i], arr[i+1] = arr[i+1], arr[i]
    else:
        for i in range(1, n - 1, 2):
            if arr[i] > arr[i+1]:
                arr[i], arr[i+1] = arr[i+1], arr[i]
```

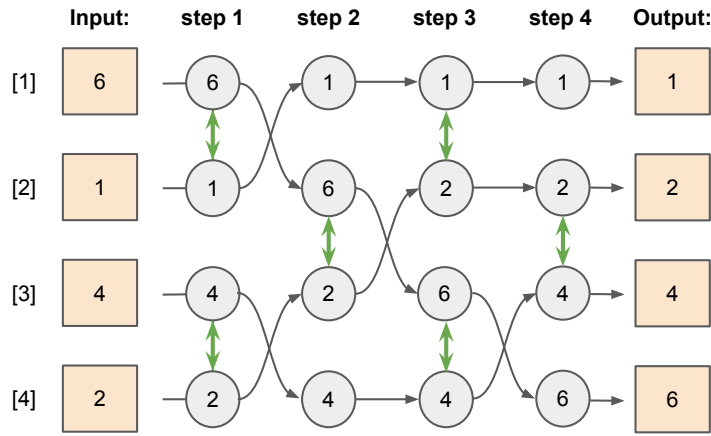


Figure A.3: An illustration of an odd-even sorting network for sorting four elements in non-descending order with an example of sorting of [6, 1, 4, 2] array.

A.5 IMPLEMENTATION DETAILS

Linear Evaluation Details. For linear evaluation, we train a linear classifier on frozen representations in a fully-supervised way, using the training set of ImageNet for training and the validation set for evaluation. We follow the training protocol of SimCLR [CKNH20] and SimSiam [CH21b] and train a linear classifier for 90 epochs using the LARS optimizer [YGG17] with the batch size of 4096, the momentum of 0.9, the linear rate of 1.6 (following the rule: learning rate = $0.1 \times \text{batch size} / 256$), without a warmup and weight decay. Following [CKNH20] and [CH21b], we use weak data augmentation (only random cropping with horizontal flipping) and apply gradient stopping on the input of the classifier to prevent updating the encoder.

SimCLR with Multiple Positives. To train SimCLR with more than one positive view per anchor, we apply contrastive loss for all possible positive pairs, considering all views from other images in the batch as negatives (with a batch of B examples with have $m(B - 1)$ negatives views). Let x_i^b denote the i 'th view of the b 'th image in a batch, and $P_{x_i^b}$ denote a set of positive samples for the anchor x_i^b , and $N_{x_i^b}$ denote a set of positive samples for the anchor x_i^b . Then,

the loss is calculated as

$$\mathcal{L}_{\text{SimCLR}} = \frac{1}{B} \sum_{b=1}^B \frac{1}{m} \sum_{i=1}^m \frac{1}{\|P_{x_i^b}\|} \sum_{y \in P_{x_i^b}} -\log \left(\frac{\exp(-d(x_i^b, y)/\tau)}{\exp(-d(x_i^b, y)/\tau) + \sum_{z \in N_{x_i^b}} \exp(-d(x_i^b, z)/\tau)} \right), \quad (\text{A.5})$$

where τ is a temperature parameter. This extension of the SimCLR framework for $m > 2$ views per image is the same as used in the SwAV evaluations [CMM⁺20]. Note that in the multi-crop scenario, we use only full-resolution global views as positive examples following “local-global” correspondence idea [CMM⁺20, CTM⁺21].

APPENDIX – EVERYTHING AT ONCE – MULTI-MODAL FUSION TRANSFORMER FOR VIDEO RETRIEVAL

This appendix contains supplementary material related to Chapter 4, *Everything at Once – Multi-modal Fusion Transformer for Video Retrieval*, of this thesis. First, we provide additional experimental results in Section B.1; then, we perform a qualitative analysis of zero-shot text-to-video retrieval in Section B.2; and finally, we provide more implementation details in Section B.3.

Contents

B.1	Additional Experimental Evaluation	217
B.2	Qualitative Analysis	219
B.3	Implementation Details	220

B.1 ADDITIONAL EXPERIMENTAL EVALUATION

CLIP Backbone. We additionally tested our model with stronger visual and text backbones. Namely, we used CLIP backbones (Contrastive Language-Image Pretraining) [RKH⁺21b] pretrained on the large Wikipedia-based image-text WiT dataset. We used the ViT-B/32 model and extracted one 512-dimensional feature per second for video and one 512-dimensional feature per word for text. For both modalities, we adopt features after projection into the multi-modal embedding space. Performance of zero-shot text-to-video retrieval and text-to-video retrieval after finetuning is presented in Table B.1. We note that using CLIP features is especially beneficial for the MSR-VTT dataset, but performance on YouCook2 also improves compared to R152 + RX101 and word2vec backbones. We also note that performance on MSR-VTT after finetuning is coming close to the performance of the CLIP4CLIP [LJZ⁺21] model that, however, is not directly comparable to ours. Compared to CLIP4CLIP, we are not finetuning backbones and we also are using a smaller MSR-VTT train subset for finetuning (7,000 clips compared to 9,000 clips).

Action Segmentation. Following [CRD⁺21] we additionally report temporal action segmentation performance on the CrossTask and Mining YouTube datasets as proposed in [KIRG19]. We measured a frame-wise video segmentation performance given the order of actions in a video. Following inference procedure [KIRG19] we computed temporal alignment of video frames based on similarity matrix to text labels by a Viterbi-decoding. Before decoding, we transferred the similarity matrix to class probabilities by applying softmax with temperature 0.05 across all labels over all videos (as we did in NCE during training). Segmentation performance is measured by an intersection over union $IoU = \frac{G \cap D}{G \cup D}$ – the ratio between the intersection of ground truth action G and prediction D and the union of them – as well as an intersection over detection $IoD = \frac{G \cap D}{D}$.

In Table B.2 we show IoU and IoD for temporal action segmentation with a recall for step action localization. We observe that our method shows a marginal boost in temporal

Method	Visual Backbone	Text Backbone	FT	YouCook2		MSR-VTT	
				R@5↑	R@10↑	R@5↑	R@10↑
Ours	R152+RX101	word2vec		40.7	51.3	23.8	31.8
Ours	CLIP	word2vec		42.7	54.0	29.0	38.7
Ours	CLIP	CLIP		42.6	54.3	32.5	42.4
CLIP4CLIP	CLIP	CLIP		-	-	57.0	66.9
Ours	R152+RX101	word2vec	✓	59.1	70.9	52.1	63.7
Ours	CLIP	word2vec	✓	62.1	72.6	60.7	72.7
Ours	CLIP	CLIP	✓	62.1	72.9	62.7	75.0
CLIP4CLIP	CLIP	CLIP	✓	-	-	70.7	80.5

Table B.1: Text-to-video retrieval on the YouCook2/MSR-VTT in zero-shot and finetune settings with CLIP backbones. As the video representation, we again use *va* – the fused video and audio modalities. FT: finetuning on downstream task. We include CLIP4CLIP [LJZ⁺21] for completeness but do not directly compare because of different pretraining and a different MSR-VTT train subset.

Method	CrossTask			Mining YouTube		
	Recall↑	IOD↑	IOU↑	Recall↑	IOD↑	IOU↑
Mining YouTube [KIRG19]	-	-	-	-	19.2	9.8
MCN [CRD ⁺ 21]	35.1	33.6	22.2	18.1	32.0	23.1
Ours	39.3	32.5	18.5	19.4	32.7	23.1

Table B.2: Evaluation of zero-shot action segmentation on the CrossTask/Mining YouTube. We report results for “R152 + RX101” visual backbone (the same as used in MCN [CRD⁺21]).

action segmentation on the Mining YouTube dataset while it does not benefit on the CrossTask dataset. However, we note that the segmentation evaluation procedure relies on the given order of steps in a video, while in the CrossTask dataset about 30% steps are missed and step orders are not always correct [ZAC⁺19b], so we consider the step action localization recall as a primary metric for this dataset, where our method improves performance by 4% with respect to MCN [CRD⁺21] baseline.

CrossTask Specific Results. To further analyze step action localization performance, we considered recalls for every specific task of the CrossTask dataset in Table B.3. We note that our method shows a significant boost in almost all cooking-related categories, like “Make Banana Ice Cream” or “Grill Steak” while does not improve performance in not-cooking categories “Change Tire,” “All Oil to Car,” and “Build a Shelves.” The MCN method, which also utilizes audio channel, similarly demonstrate a lower performance in “Change Tire” and “All Oil to Car” tasks compared to video-text-only the CrossTask [ZAC⁺19b] and HT100M [MZA⁺19] baselines. We can assume that this happens due to the fewer car-related video clips in the HowTo100M dataset (7.8M) compared to food-related clips (54.4 M).

Fusion Transformer Ablation. We also ablate our Fusion Transformer with respect to the number of transformer blocks and the number of heads of multi-headed attention in Table B.4. Due to resource constraints, for an increase in the number of transformer blocks, we should linearly decrease either the number of heads or the training batch size (the large batch size is essential due to contrastive training). We observe that the best performing configuration consists of 1 transformer block and a maximum number of transformer heads (64 heads) that fits into resources, however, we assume the model can further boost performance by increasing the number of transformer blocks leveraging more resources.

Method	Make Kimchi Rice	Pickle Cucumber	Make Banana Ice Cream	Grill Steak	Jack Up Car	Make Jello Shots	Change Tire	Make Lemonade	Add Oil to Car	Make Latte	Build Shelves	Make Taco Salad	Make French Toast	Make Irish Coffee	Make Strawberry Cake	Make Pancakes	Make Meringue	Make Fish Curry	Average
Fully-supervised baseline [ZAC ⁺ 19b]	19.1	25.3	38.0	37.5	25.7	28.2	54.3	25.8	18.3	31.2	47.7	12.0	39.5	23.4	30.9	41.1	53.4	17.3	31.6
CrossTask [ZAC ⁺ 19b]	13.3	18.0	23.4	23.1	16.9	16.5	30.7	21.6	4.6	19.5	35.3	10.0	32.3	13.8	29.5	37.6	43.0	13.3	22.4
HT100M et al. [MZA ⁺ 19]	33.5	27.1	36.6	37.9	24.1	35.6	32.7	35.1	30.7	28.5	43.2	19.8	34.7	33.6	40.4	41.6	41.9	27.4	33.6
MCN	25.5	31.1	39.7	32.7	35.4	36.8	29.0	40.0	28.4	33.8	45.7	27.5	36.1	34.9	39.6	42.6	43.0	29.1	35.1
Ours	30.5	41.2	46.5	46.6	38.9	32.0	19.5	48.9	25.8	33.6	44.7	29.1	40.7	36.9	50.7	44.1	63.1	33.6	39.2

Table B.3: Step action localization performance on the CrossTask [ZAC⁺19b] dataset: recalls corresponding to every specific task.

#blocks	#heads	hidden s.	batch size	YouCook2		MSR-VTT	
				R@5↑	R@10↑	R@5↑	R@10↑
1	64	4096	224 × 10	40.7	51.3	23.8	31.8
2	64	4096	224 × 5	37.3	47.6	23.2	32.5
2	64	4096	112 × 10	38.6	49.8	20.8	28.6
2	32	2048	224 × 10	38.1	49.1	22.6	30.9
4	16	1024	224 × 10	35.4	46.8	23.7	31.7

Table B.4: Evaluation of different fusion transformer architectures. *#blocks* stands for a number of transformer blocks, *#heads* – for a count of attention heads; *hidden s.* denotes a hidden size of the transformer layers (that linearly depends on the number of heads); *batch size* denotes a training batch size where $x \times y$ means that we use a batch of x videos and randomly sample y clips per video.

Text-Audio Disentangling and Loss Weighting. We also show the importance of disentangling audio with respect to text while training on the HowTo100M dataset, as well as the importance of using a larger text-video loss weight in the loss function in Table B.5. Since text is obtained by applying an ASR system to the audio track, to avoid text being learned just as an audio narration, we shift the audio clip randomly by half of clip length (4 seconds out of 8 seconds) with respect to the video and text boundaries in “*disentangled text-audio*.” To further regularize text-audio learning, in “*+ loss weighting*” we used a larger weight for a text-visual loss $\lambda_{t_v} = 1$ compared to other loss components $\lambda_{v_a} = \lambda_{t_a} = \lambda_{t_{va}} = \lambda_{v_{ta}} = \lambda_{a_{tv}} = 0.1$ (similarly to [ARS⁺20]). Table B.5 shows that both adaptations are beneficial for training on the HowTo100M dataset.

Relative Positional Encodings. As demonstrated in the paper, we found that absolute positional embeddings are not beneficial for our model. Apart from them, we also tested relative positional encodings, namely the Rotary Position Embedding (RoPE) [SLP⁺21], that are shown to better generalize to longer inputs at test time. We incorporated RoPE into our attention block, where we independently apply RoPE to each sequence of text, video, or audio tokens. However, as presented in Table B.6 we also found that RoPE does not benefit our model. But a more complex strategy that e.g., adds information about alignment tokens from different modalities (similarly to the RoPE 2D case) may lead to performance improvement.

B.2 QUALITATIVE ANALYSIS

We also qualitatively analyze the zero-shot retrieval capacity of our model on the YouCook2 dataset in Figure B.2. We observe, that for all shown examples retrieved clips are semantically

Configuration	YouCook2		MSR-VTT	
	R@5↑	R@10↑	R@5↑	R@10↑
aligned text-audio	37.8	47.2	14.0	20.0
disentangled text-audio	37.2	45.7	17.9	25.0
disentangled text-audio + loss weighting	40.7	51.3	23.8	31.8

Table B.5: Evaluation of disentangling of audio and text while training on the HowTo100M dataset as well weighting components in the loss function. *Aligned text-audio* and *disentangled text-audio* were trained without loss weighting, *disentangled text-audio + loss weighting* – with $\lambda_{t_v} = 1$, $\lambda_{v_a} = \lambda_{t_a} = \lambda_{t_{va}} = \lambda_{v_{ta}} = \lambda_{a_{tv}} = 0.1$ as proposed.

Configuration	YouCook2		MSR-VTT	
	R@5↑	R@10↑	R@5↑	R@10↑
RoPE [SLP ⁺ 21]	40.4	51.2	23.1	31.6
no positional emb.	40.7	51.3	23.8	31.8

Table B.6: Evaluation of Rotary Position Embedding (RoPE).

related to the given text query. Even when a correct video does not occur in the top-5 retrieval results, top-5 videos correspond to the text input: for example, for a query “bring a large pan of water to boil” our model predicts videos with boiling water in a pot.

B.3 IMPLEMENTATION DETAILS

Audio Backbone. Following [RBH⁺21, CRD⁺21], as an audio backbone, we use a trainable CNN with residual layers adopted from [HRS⁺18] that takes log-mel spectrograms with 16 kHz sampling rate, 25 ms Hamming window, 10 ms window stride, and 40 Mel filter bands. Note that this backbone is not pretrained. Since architecture used in [RBH⁺21, CRD⁺21] extracts 6 1024-dimensional features per second, we adapt the last two residual blocks to extract ~ 1.5 4096-dimensional features per second (the same as our video backbone). We illustrate architecture in Figure B.1. While training on 8-seconds clips, we used 7.7 seconds of audio, that results exactly in 12 audio tokens.

Ablation Architectures. In Figure B.3 we illustrate 4 architectures considered in our ablation studies: a) *no transformers*: our architecture without transformer layer, trained with three pairwise contrastive losses; 2) *single modality transformer*: leveraging three separate modality-specific transformers; 3) *fusion transformer*: the proposed modality agnostic transformer, but trained with three pairwise contrastive losses without fused modality components; 4) *fusion*

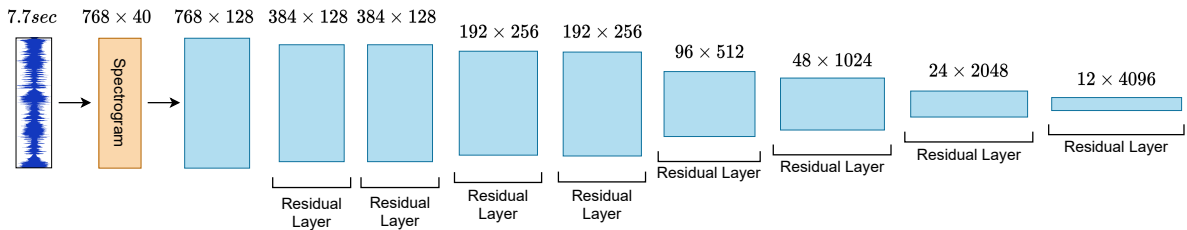


Figure B.1: The schematic visualization of audio backbone network (the illustration is inspired by [RBH⁺21]).

transformer + combinatorial loss: the proposed architecture that utilises the modality agnostic transformer with combinatorial input, trained with combinatorial loss.

finetuning Details. During finetuning on the YouCook2 and MST-VTT datasets, we set $\lambda_{t_v} = \lambda_{v_a} = \lambda_{t_a} = \lambda_{t_{va}} = \lambda_{v_{ta}} = \lambda_{a_{tv}} = 1$, and train the model for 5 epochs with a learning rate of $1e^{-5}$ and a batch size of 256 on the YouCook2 dataset, and for 25 epochs with the learning rate of $5e^{-5}$ and the batch size of 128 on the MSR-VTT dataset.

Training Time Training our model on the HowTo100M dataset takes approximately 2 days on four Nvidia V100 32GB GPUs. finetuning on the YouCook2 and the MSR-VTT takes less than 30 minutes.



Figure B.2: Qualitative evaluation. Examples of zero-shot text-to-video retrieval on the YouCook2. Each row shows the top-5 retrieved videos for a given text query. The correct video is highlighted with a red color.

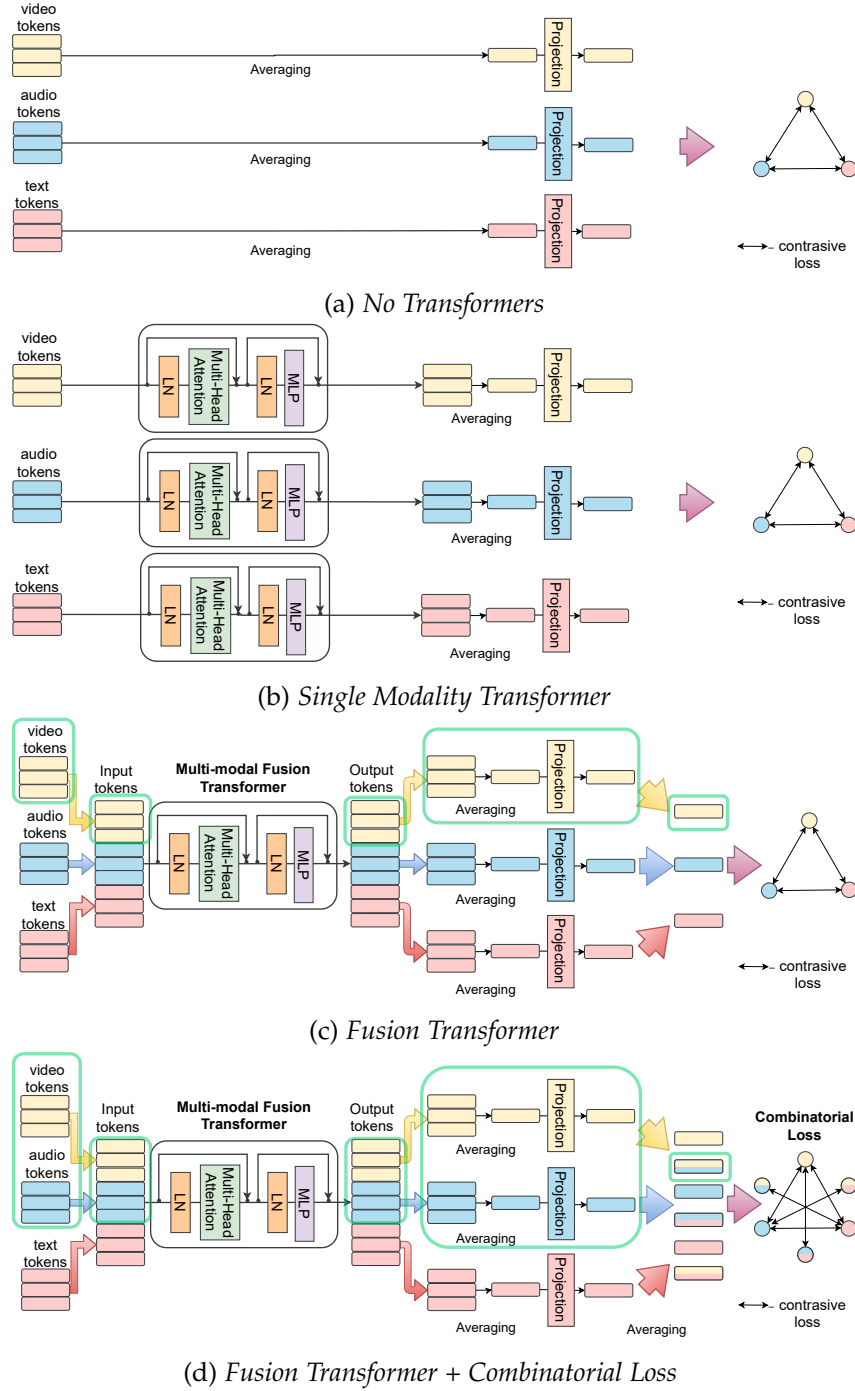


Figure B.3: Comparison of different architectures considered in ablation studies. Note that in illustration of the fusion transformer in (c) and (d), not all blocks are always active, using green rectangles we consider the video embedding computation in (c) and video-audio embedding computation in (d).

APPENDIX – MATCH, EXPAND AND IMPROVE: UNSUPERVISED FINETUNING FOR ZERO-SHOT ACTION RECOGNITION WITH LANGUAGE KNOWLEDGE

This appendix contains supplementary material related to Chapter 5, *Match, Expand and Improve: Unsupervised Finetuning for Zero-Shot Action Recognition with Language Knowledge*, of this thesis. First, we introduce dataset statistics in Section C.1), then, we provide more implementation details in Section C.2, and finally, we provide additional results in Section C.3.

Contents

C.1	Dataset Statistics	225
C.2	Implementation Details	226
C.3	Additional Results	228
C.3.1	Attention Heatmaps	228
C.3.2	Robustness Against Noisy Action Dictionary	230
C.3.3	Examples of Language Sources	231
C.3.4	Parameter-Free Temporal Module	232

C.1 DATASET STATISTICS

Kinetics-400 (K400) [KCS⁺17] is the most popular benchmark for action recognition tasks, containing around 240K training videos in 400 classes. The dataset consists of YouTube videos with an average length of 10 seconds. We use the training set of K400 for finetuning CLIP.

UCF101 [SZS12] is collected from YouTube videos, consisting of 13K videos from 101 classes. There are three splits of training data (~ 9.4 K) and validation data (~ 3.6 K). Following XL-CIP [NPC⁺22] and ViFi-CLIP [RKM⁺22], we report the average performance on the three validation splits.

HMDB51 [KJG⁺11] consists of 7K videos comprised of 51 action classes, collected from YouTube videos and movie clips. There are three splits of training data (~ 3.5 K, 70 videos per class) and validation data (~ 1.5 K, 30 videos per class). Following [NPC⁺22, RKM⁺22], we report the average performance on the three validation splits.

Kinetics-600 (K600) [CNBH⁺18] is an extension of K400, consisting of 650K videos in 600 classes. Following [CH21a, NPC⁺22, RKM⁺22], we use the 220 classes¹ that are not included in K400 for zero-shot action recognition. There are three validation splits, each containing 160 classes randomly sampled from these 220 classes. We report the average performance on the three validation splits, each containing around 14K videos.

MiniSSv2 [CPR⁺21] (87 classes, 93K videos) is a subset of Something-Something v2 (SSv2) [GEKM⁺17a] (174 classes, 220K videos). SSv2 is an egocentric motion-based action

¹In the evolution from K400 to K600, there are renamed, removed and split classes. See details in Appendix B in [CH21a].



Figure C.1: Attention heatmaps on actions which have a verb form (lemma or gerund) directly included in the K400 dictionary. We compare among CLIP (2nd row), ViFi-CLIP (3rd row) and our MAXI (4th row). Warm and cold colors indicate high and low attention. MAXI has more focused attention on hands (for *clap*) and legs (for *kick ball*).

dataset, which has a large visual domain shift to K400. Furthermore, the action classes are detailed descriptions of fine-grained movements, in a largely different language style than the K400 action dictionary, e.g. *Failing to put something into something because something does not fit*, and *Lifting a surface with something on it but not enough for it to slide down*. For zero-shot action recognition, we evaluate on the validation split of MiniSSv2 (12K videos). For few-shot action recognition, we follow [RKM⁺22] and evaluate on the validation split of SSv2 (25K videos).

Charades [SVW⁺16] is a long-range activity dataset recorded by people in their homes based on provided scripts for home activities. There are ~ 10 K videos in 157 classes. The average video length is 30 seconds. Each video has annotations of an average of 6.8 action instances, often in complex co-occurring cases. The validation split consists of 1.8K videos. We report the mean Average Precision (mAP) for the multi-label classification task.

Moments-in-Time (MiT) [MAZ⁺19] is a large-scale action dataset of 3-second YouTube video clips, which cover actions in 305 classes, performed by both humans and animals. The validation split consists of 30K videos.

UAV Human (UAV) [LLZ⁺21b] is an action dataset recorded with an Unmanned Aerial Vehicle in unique camera viewpoints. There are 155 action classes. Actions in different categories are performed by a fixed group of subjects in the same background scenes. This leads to an extremely low object-scene bias and a large shift to the domain of K400 and CLIP. We evaluate on the RGB videos and report the average performance on the two official validation splits, each consisting of ~ 6.2 K videos.

C.2 IMPLEMENTATION DETAILS

In addition to the details mentioned in the main part of thesis, we cover more implementation specifics here.

CLIP matching. The CLIP matching step is for consuming the language source of the predefined action dictionary D . We use CLIP² [RKH⁺21a] with the ViT-B/16 visual encoder [DBK⁺21a] to match each video with texts in the predefined action dictionary. To improve the matching

²CLIP model source

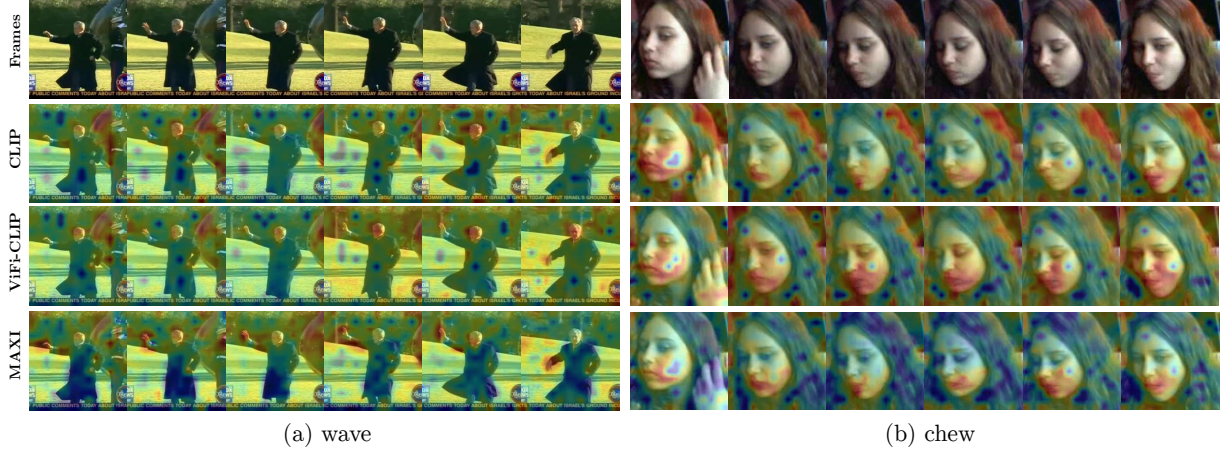


Figure C.2: Attention heatmaps on novel actions which do not have any verb form included in the K400 dictionary. We compare among CLIP (2nd row), ViFi-CLIP (3rd row) and our MAXI (4th row). Warm and cold colors indicate high and low attention. MAXI has more focused attention on hand and arm for *wave*, and on the area of mouth for *chew*.

quality for Text Bag Construction, we perform prompt ensembling over the 28 prompt templates³ which are proposed by CLIP for Kinetics videos. Important to note, during inference we follow the exact protocol of ViFi-CLIP [RKM⁺22] and use only a single prompt.

GPT-3 text expansion. We employ the GPT-3 text-davinci-003 model [BMR⁺20]. We set the temperature to 0.4. We generate 5 verb phrases using the input instruction - *Generate 5 phrases to describe the action of $\langle action \rangle$ in simple words.* Here for a video x_i , $\langle action \rangle$ is the best matched text $\hat{t}_i$ from the predefined action dictionary.

BLIP captioning. We use BLIP model⁴ [LLXH22a] with ViT-L/16 as the image captioner. For each video, the image captioning is performed on 8 uniformly sampled frames. The frames are resized into 384.

Text augmentation. We use the natural language processing tool spaCy⁵ to parse the verbs and verb phrases from the descriptions. We perform augmentation by converting the verbs into forms of lemma and gerund (present participle) and include results in the text bag.

Training. We employ CLIP with the ViT-B/16 visual encoder. We follow the full-finetuning configuration of [RKM⁺22] to finetune both the visual and text encoder. During training, we follow the configuration of [RKM⁺22, NPC⁺22] for visual augmentation of multi scale crop, random flipping, color jittering and gray scaling. We do not perform augmentations of MixUp or CutMix.

As different videos have varying numbers of texts in their bags, we randomly sample N_{bag} texts from the originally constructed bag in each training iteration. For multiple instance learning, we use all the N_{bag} words in a text bag to form N_{bag} text prompts for each video. The text prompt is in the format of $\langle text1 \rangle + \langle text2 \rangle$. The first part $\langle text1 \rangle$ is uniform for all the N_{bag} text prompts. Specifically, we use a hand-crafted prompt template *a photo of $\langle action \rangle$* , where $\langle action \rangle$ is the best-matched text $\hat{t}_i$ from the predefined action dictionary (see Equation (5.1) in the main part of the thesis). $\langle text2 \rangle$ is an individual text from the text bag. To avoid duplication, we do not use $\hat{t}_i$ as $\langle text2 \rangle$.

Inference. We follow [NPC⁺22, RKM⁺22] and sample a single view via sparse temporal

³<https://github.com/openai/CLIP/blob/main/data/prompts.md>

⁴BLIP model source

⁵spaCy <https://spacy.io/>

sampling and spatial center crop. The same single prompt template is used in inference.

C.3 ADDITIONAL RESULTS

C.3.1 Attention Heatmaps

To gain more insights into the performance improvement of MAXI, we compare the visualizations of attention heatmaps across several approaches in Figure C.1, Figure C.2 and Figure C.3. CLIP is the original CLIP [RKH⁺21a] without any finetuning. ViFi-CLIP [RKM⁺22] finetunes CLIP via supervised classification on K400 with ground truth annotations. MAXI is our approach of unsupervised finetuning with language knowledge.

We obtain the attention maps by computing the cosine similarity between the patch token features from the visual encoder and the text feature from the text encoder. We visualize the attention maps in several action classes from the downstream datasets used for the zero-shot action recognition task. Based on the relationship between the zero-shot action class and the K400 action dictionary used for training, we categorize the visualizations into 3 groups: (1) In-dictionary action classes which have a verb form (lemma or gerund) directly included in the K400 action dictionary, e.g. *clap* and *kick ball* in Figure C.1; (2) Novel actions classes which do not have any verb form included in the K400 action dictionary, e.g. *wave* and *chew* in Figure C.2; (3) General actions whose verb form is a basic component of several actions in the K400 action dictionary, e.g. *catch*, *hit*, *jump* and *run* in Figure C.3.

In-dictionary action classes. In Figure C.1, we visualize two samples of the action *clap* and *kick ball*. *clap* has the same lemma as *clapping* in the K400 dictionary, while *kick ball* has related actions of *kicking field goal* and *kicking soccer ball* in the K400 dictionary. We see that CLIP has incorrectly high attention on object (Figure C.1(a), 2nd row) or background scene (Figure C.1(b), 2nd row). ViFi-CLIP has cluttered high attention on both the subjects and the background scenes. On the contrary, MAXI has more focused attention on the hands (for *clap*) and legs (for *kick ball*).

In our GPT-3 text bag of *clapping*, related words such as *clap*, *smacking hands*, *slapping palms* and *clapping hands* are included. This strengthens the association between the action *clap* and the body part of hands, and leads to more accurate attention. Furthermore, in BLIP caption verb text bags, the verb *clap* appears several times in frame captions of K400 videos of *clapping*, *giving or receiving award* and *applauding*. This further improves the understanding of *clap*. Similarly, in BLIP frame captions, *kick* is an even more basic verb with large amount of occurrences.

Novel action classes. In Figure C.2, we compare the attention maps for the novel verbs *wave* and *chew* that do not appear in the K400 action dictionary. We see that for *wave*, CLIP and ViFi-CLIP have attention on the background scene or on the head, while MAXI has correct attention on the hand and arm. For *chew*, CLIP has more attention on the hair and ViFi-CLIP has attention on a large area of the face. On the contrary, MAXI has consistent focused attentions on the area of the mouth where the action *chew* happens.

The verb *wave* appears in BLIP caption verb text bags of several K400 videos of *clapping*, *applauding*, *celebrating*. The verb *chew* appears in captions of K400 videos of *eating carrots*, *eating spaghetti*, *eating watermelon* and *baby waking up*. The additional language source improves the knowledge of actions that never appear in the K400 action dictionary.

General actions. In Figure C.3, we illustrate the attention maps for four general verbs *catch*, *hit*, *jump* and *run*. These verbs are basic components of several actions in the K400 dictionary,

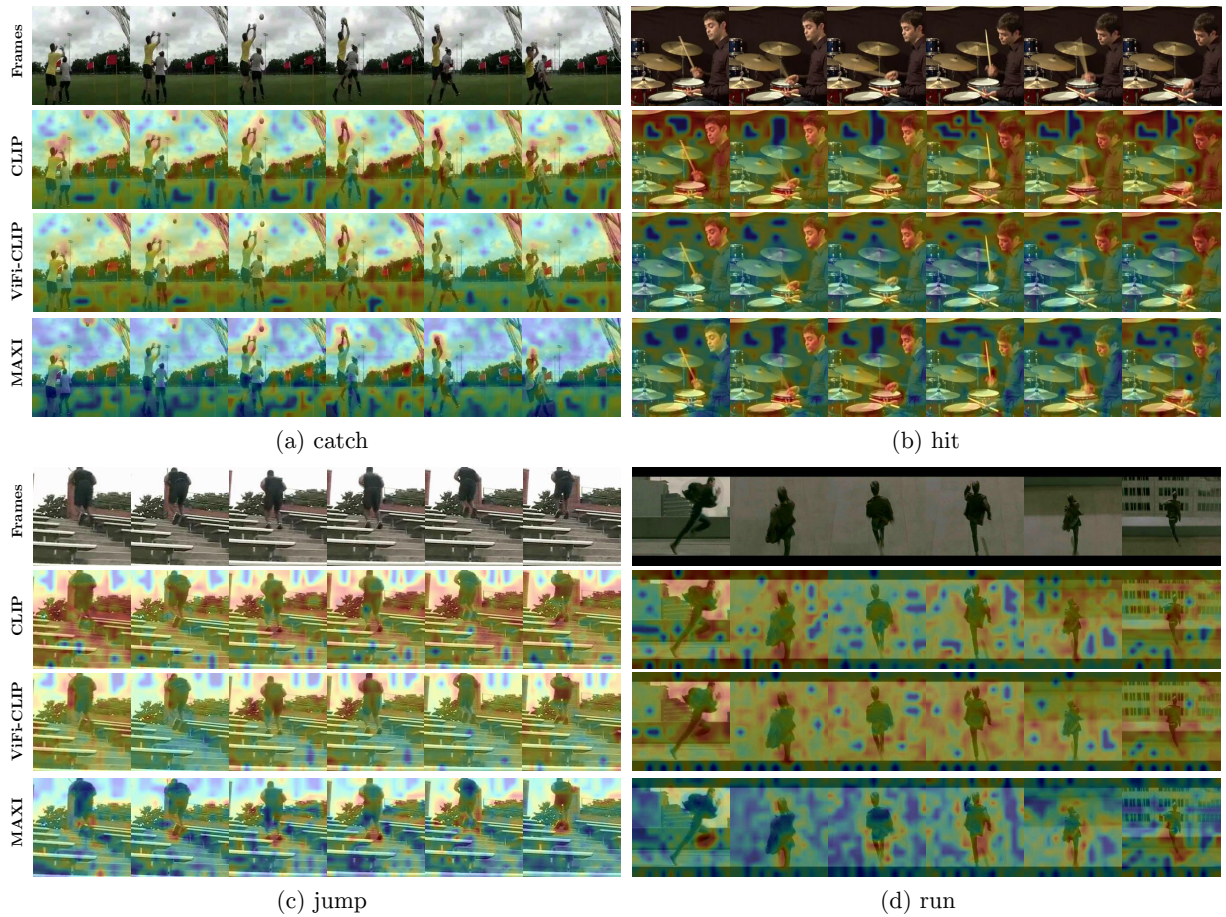


Figure C.3: Attention heatmaps on actions which have a verb form (lemma or gerund) directly included in the K400 dictionary. We compare among CLIP (2nd row), ViFi-CLIP (3rd row) and our MAXI (4th row). Warm and cold colors indicate high and low attention. MAXI has more concentrated attention on the part where the action happens, e.g. catching ball with hands (Figure C.3(a), 4th row), hitting drum with stick (Figure C.3(b), 4th row), legs and feet jump on stairs (Figure C.3(c), 4th row), and attention on the running body (Figure C.3(d), 4th row).

Action dictionary	Dictionary size	UCF101	HMDB51	K600	MiniSSv2	Charades	UAV Human	Moments-in-time
CLIP(w/o finetune)	Zero-Shot	69.93 / 92.7	38.02 / 66.34	63.48 / 86.80	3.96 / 14.42	19.80	1.79 / 7.05	20.11 / 40.81
K400	400	78.18 / 96.03	50.35 / 77.10	70.78 / 92.17	5.74 / 17.70	23.89	3.06 / 9.46	22.41 / 45.83
K400+WebVid2.5M	800	75.99 / 96.00	<u>45.97</u> / 73.94	69.14 / 91.13	<u>4.81</u> / <u>15.79</u>	22.67	2.11 / 8.00	20.92 / 43.99
K400+WebVid2.5M	1200	75.72 / 96.02	<u>45.51</u> / <u>73.97</u>	69.36 / 91.11	<u>4.21</u> / 15.15	22.35	<u>2.39</u> / 7.98	<u>21.29</u> / <u>44.33</u>
K400+WebVid2.5M	1600	<u>76.14</u> / 96.01	44.84 / 71.79	69.23 / 91.10	4.42 / 14.71	<u>22.89</u>	2.14 / 7.71	20.69 / 43.59

Table C.1: Robustness of finetuning with noisy action dictionaries. We add noisy verbs parsed from the WebVid2.5M dataset into the original K400 action dictionary. We report the zero-shot transfer performance (mAP on Charades and Top1/Top5 accuracy on other datasets). We set the text bag filtering ratio $p = 50\%$ for improved text bag quality.

Temp. attention layers	UCF101	HMDB51	K600	MiniSSv2	Charades	UAV Human	Moments-in-time
None	78.17	52.24	71.43	6.37	23.79	<u>2.72</u>	22.91
2	<u>77.38</u>	51.83	<u>70.41</u>	5.98	<u>22.87</u>	2.90	22.51
6	75.91	<u>51.92</u>	69.23	<u>6.09</u>	21.78	2.52	<u>22.52</u>

Table C.2: Cross-frame temporal attention modules. we report the zero-shot transfer performance after finetuning CLIP on K400. We train with text bags of GPT3 verbs and BLIP verbs. We set the text bag filtering ratio $p = 90\%$. Adding temporal attention module does not lead to performance improvement.

e.g. *catching fish*, *catching or throwing frisbee*, *hitting baseball*, *jumping into pool* and *running on treadmill*. In these samples, CLIP and ViFi-CLIP have cluttered attention on the background scene or objects. MAXI has more concentrated attention on the part where the action happens, e.g. *catching ball with hands* (Figure C.3(a), last row), *hitting drum with stick* (Figure C.3(b), last row), *legs and feet jump on stairs* (Figure C.3(c), last row), and attention on the running body (Figure C.3(d), last row).

These verbs are very general and could have highly diverse instantiations. E.g. *hit* (drum) in Figure C.3(b) is not close to *hitting baseball* on K400. *jump* (on stairs) in Figure C.3(c) is not close to *jumping into pool* or *bungee jumping* on K400, even if they share the same verb. In our GPT-3 verb bag and BLIP caption verb bag, there is a large amount of these verb instances that facilitate the comprehensive understanding of these general verbs. This leads to better focus even in unusual complex scenes, e.g. jumping on stairs (Figure C.3(c)).

C.3.2 Robustness Against Noisy Action Dictionary

In Table 5.5 in the main part of the thesis, we explored the robustness of our finetuning pipeline against noisy action dictionaries. In case of an over-specified dictionary, we added noisy verbs and verb phrases into the original K400 action dictionary. The noisy verbs are parsed from the captions in the WebVid2.5M dataset [BNVZ21a]. Here we further increase the ratio of noisy verbs, and add 800 and 1200 verbs into the dictionary, resulting in 1200-class and 1600-class spaces.

In Table C.1, we report the zero-shot transfer performance of models finetuned with the resulted 1200-class and 1600-class space. We set the text bag filtering ratio $p = 50\%$ for improved text bag quality. We see that even with extremely noisy dictionary where 50% to 75% of words do not match with the video data, our finetuning still results in a robust zero-shot transfer performance to unseen datasets. The robustness is the consequence of the fact that we collect knowledge from multiple language sources and learn from them via Multiple Instance

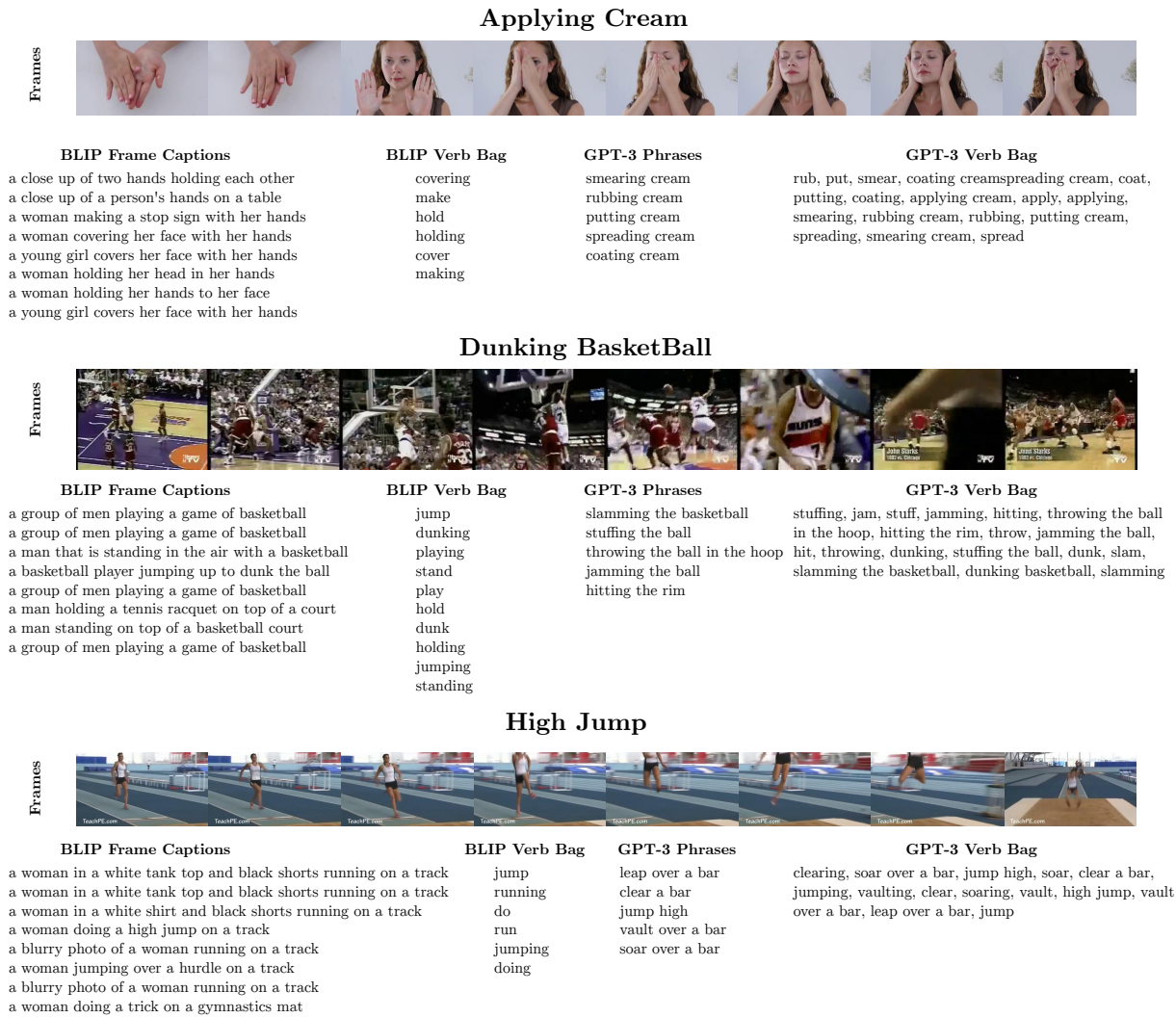


Figure C.4: Examples of video frames, BLIP frame captions, GPT-3 phrases, together with the derived BLIP verb bag and GPT-3 verb bag. The videos are from the K400 dataset.

Learning. Note that the zero-shot transfer does not have consistent change in performance across the downstream datasets, as different datasets have different language domain shift to the action dictionary used for training.

C.3.3 Examples of Language Sources

Similar to the *cooking egg* example in Figure 5.2 in the main part of the thesis, we illustrate more examples of video frames, BLIP frame captions, GPT-3 phrases, together with the derived BLIP verb bag and GPT-3 verb bag in Figure C.4. The videos are from the unlabeled K400 dataset which we use for training.

C.3.4 Parameter-Free Temporal Module

As mentioned in Section 5.3.1 in the main part of the thesis, we explore a parameter-free temporal-aware module on the CLIP model. We modify the multi head attention module [VSP⁺17] in the visual encoder of CLIP to be temporal aware. Originally, the attention on the frame t is computed via $A_t(Q_t, K_t, V_t) = \text{softmax}_{\frac{Q_t K_t^\top}{d_k}} V_t$, where Q_t , K_t and V_t are the query, key and value from frame t .

We explore to compute the cross-frame attention via

$$A'_t(Q_t, K_{t+i}|_{i \in I}, V_t) = \text{softmax}_{\frac{\sum_{i \in I} (Q_t \cdot K_{t+i}^\top) / |I|}{d_k}} V_t \quad (\text{C.1})$$

where we set $I = \{-1, 0, 1\}$. In this case, we use the keys from the frame $t - 1$, t and $t + 1$ to compute the attention for frame t .

We apply the cross-frame attention on the last 2 and on the last 6 transformer layers in the visual encoder of CLIP. In Table C.2, we report the zero-shot transfer performance. We see that in comparison to the variant without any temporal attention module, using cross-frame attention does not lead to performance improvement. K₄₀₀ is of far smaller scale in comparison to the original CLIP domain. Finetuning from the CLIP model weights with a modified architecture could result in the case that the model drifts far away from the wise CLIP source domain. The results are consistent with the claims in [RKM⁺22] that a sophisticated temporal module does not necessarily lead to performance improvement.

APPENDIX – IN-STYLE: BRIDGING TEXT AND UNCURATED VIDEOS WITH STYLE TRANSFER FOR TEXT-VIDEO RETRIEVAL

This appendix contains supplementary material related to Chapter 6, *In-Style: Bridging Text and Uncurated Videos with Style Transfer for Text-Video Retrieval*, of this thesis. We first elaborate on some discussions in Section D.1; further, we provide In-Style Method details in Section D.2 and implementation details in Section D.3; and finally, we provide more qualitative evaluations in Section D.4 and discuss limitations in Section D.5.

Contents

D.1	Additional Discussions	233
D.2	In-Style Method Details	236
D.3	Implementation Details	237
D.4	Qualitative Results	238
D.5	Limitations	238

D.1 ADDITIONAL DISCUSSIONS

Text Query Style. Figure D.1 and Figure D.2 show the respective word clouds for the five datasets with and without stop words. In Table D.1, we show five different text examples from the different datasets considered in this paper to further highlight the differences in the text styles.

Model Generalization. In the following, we discuss the generalization performance of the model trained on different text styles (Table 6.2 in the main part of the thesis.) When we consider models trained only with one text style (text queries that are only from one dataset) in the top half of Table 6.2, it shows that the mean retrieval performance is higher for the queries with the style of MSR-VTT, DiDeMo, or LSMDC datasets. Interestingly, MSVD-style text queries, which are similar to MSR-VTT queries in terms of sentence structure (Table D.1) and usage of stop words (Figure D.1 and Figure D.2), show similar MSR-VTT retrieval performance compared to DiDeMo and LSMDC-style text queries, and moreover, lower performance on the DiDeMo, and LSMDC datasets. We hypothesize that longer and more descriptive text is beneficial for model generalization (the MSVD dataset contains the shortest text descriptions of all datasets).

In Table D.2a, we also report the number of shared video clips in generated pairs P_{gen} with different datasets' text styles. Interestingly, P_{gen} based on the LSMDC text queries has the smallest number of pairs in general. Moreover, even though it has low overlap in videos with P_{gen} from all other datasets (0.18–0.24, see in Table D.2b), LSMDC's P_{gen} shows one of the highest generalization to the MSVD, DiDeMo, and MSR-VTT datasets. Analyzing examples in Table D.1, we suggest that LSMDC captions are more concise and descriptive in terms of

Dataset	Examples
MSR-VTT (~43 symbols in a text)	1) A bulldozer removes dirt 2) An infomercial with a pharmaceutical company talking about an epilepsy drug pending approval from the FDA 3) Extreme violence scenes with people fighting with each other 4) A woman is in front of a whiteboard talking about the numbers written on it 5) A man is playing an instrument
YouCook2 (~39 symbols in a text)	1) Add some herb sprinkle and stir the meat 2) Bake the pizza on the grill 3) Pour butter into the wok 4) Peel an onion and chop into pieces 5) Add the tomato paste crushed tomatoes tomato puree and beef stock to the pan
DiDeMo (~147 symbols in a text)	1) First time we see the dancers go down on one leg the men hit the ground with their sticks. They first start crouching and hitting the ground with the sticks 2) When the man puts his head down the guitar player is looking up. The guitarist is looking straight up. A man plays the guitar while looking up. The guitarist is looking straight up as he plays. 3) Red phone booth is visible a red phone booth is in the scene. A person walks in the middle of the camera. A red phone booth can be seen a red telephone booth is on the sidewalk. 4) The camera moves back to the left to the tree. White square exits frame left the camera pans back the way it came. Square area lines with stones comes into view the fence comes into view. 5) Fog moves in toward the ice skater. a woman spins around several times very fast. A woman pirouettes as she comes near the camera. Woman spins more than 5 times in a row.
MSVD (~31 symbols in a text)	1) A lady is pouring raw strawberry juice into a bowl 2) A man is slicing the crust into a potato 3) A man lifts three sunflowers 4) A man is putting a pan into an oven 5) A boy rides around in circles on a tricycle
LSMDC (~46 symbols in a text)	1) The dish is covered in saffron and spices. 2) He slaps SOMEONE again. 3) He and SOMEONE join forces to grab the cube, which's connected with several more wire. 4) In the race, a rider falls. 5) SOMEONE dashes to a clothes closet and ducks inside. The cup spins across the floor.
VATEX (~71 symbols in a text)	1) Someone is demonstrating how to paint a metal sheet on a window 2) A person is cooking scallops in the pan over a fire place and then begins to pour them in some water 3) Two women make a video tutorial on how to bake cookies 4) A person is throwing garbage into the trash can and talking 5) A woman is outside and preparing an edible meal by inserting herbs into it, then placing them on the ground
Food.com (~54 symbols in a text)	1) In a blender or food processor, puree the first 3 ingredients until smooth 2) Combine juice, remaining 2 tablespoons sugar and lemon juice 3) Place chicken on wire rack 4) Then toss in the artichokes and serve immediately 5) Cut beef into 3 - inch pieces
WikiHow (~41 symbols in a text)	1) Do lunges in the park 2) Cut a large milk jug 3) Buy your favorite knit kit 4) Trim the lining of your sweater 5) If your child is in a fight, put his hands on the sand

Table D.1: Five random examples of text descriptions in different datasets. With the dataset name, we also report the median length of a text in the dataset.

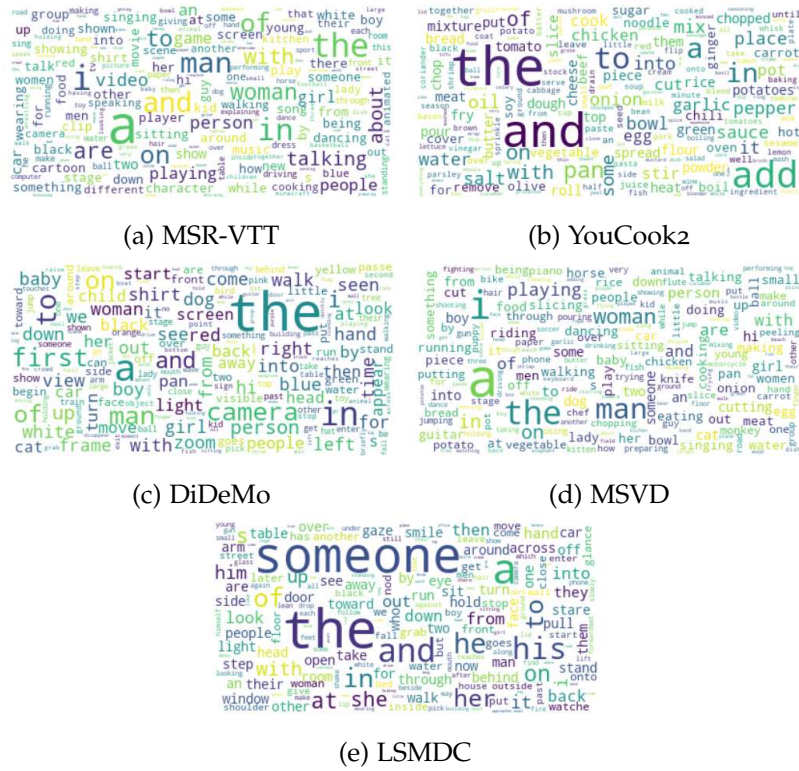
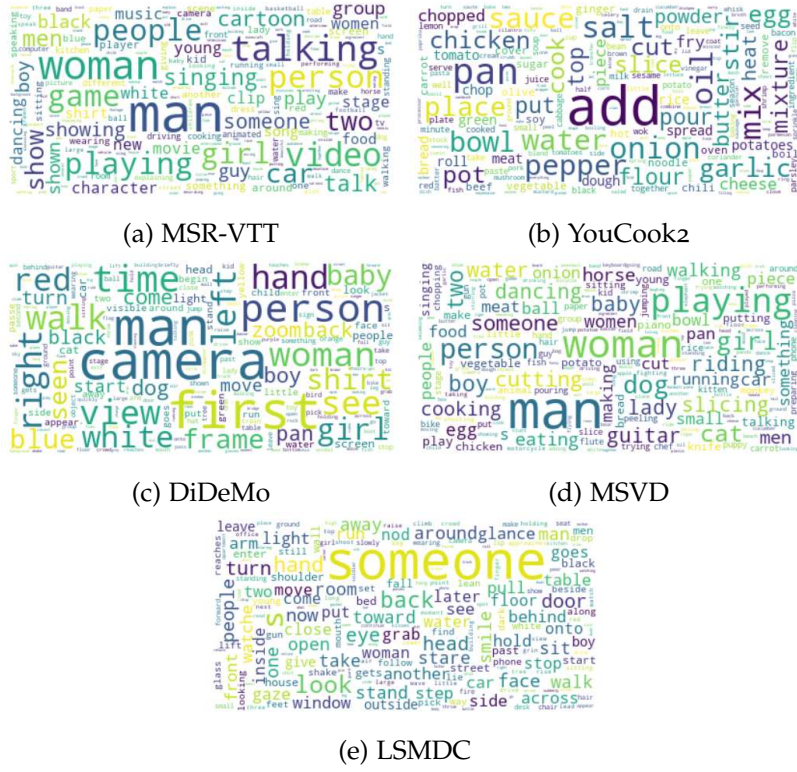


Figure D.1: Word clouds for text queries from different datasets (stop words are **included**).

object-verb-subject details (who does what in a video). For example, in the 3rd MSR-VTT example, there is only a general description that people are fighting without specification of exact people and their actions, and in the 5th example, the caption only says that a man is playing an instrument, without specification of an instrument. At the same time, in LSMDC texts, the object-action-subject description is more detailed, such as in the first example, all consecutive actions are specified, or in the 2nd example, the exact hitting action (“he slaps”) is specified. Therefore, we hypothesize that a text style with more concise descriptions is better for model generalization. However, as Table 2 demonstrates, the best mean performance over all datasets is achieved while training with different target text styles.

Figure D.2: Word clouds for text queries from different datasets (stop words are **excluded**).

D.2 IN-STYLE METHOD DETAILS

Captioner. For the captioner, we follow BLIP [LLXH22b] image captioner architecture, which we extend to video captioning as shown in Figure D.3a. Namely, we encode m uniformly sampled frames (we use $m = 8$ for training and $m = 12$ for inference) from a video by the image transformer to obtain frame-wise tokens. Then, we feed this set of encoded tokens from all the frames into the cross-attention of an image-grounded text decoder. Therefore, the predicted text is conditioned on multiple video frames at once. The image-grounded text decoder predicts the next text token given an input of previous text tokens (where the “dec” token is concatenated to the beginning of the input sequence and denotes the start of the output). During the inference, the text tokens are generated one by one in an autoregressive manner.

Dataset	MSR-VTT	YouCook2	DiDeMo	MSVD	LSMDC
MSR-VTT	495k	111k	160k	239k	87k
YouCook	111k	168k	65k	97k	40k
Didemo	160k	65k	280k	135k	57k
MSVD	239k	97k	135k	379k	75k
LSMDC	87k	40k	57k	75k	144k

(a) Number of shared video clips

Dataset	MSR-VTT	YouCook2	DiDeMo	MSVD	LSMDC
MSR-VTT	1	0.23	0.32	0.48	0.18
YouCook2	0.66	1	0.39	0.58	0.24
DiDeMo	0.57	0.23	1	0.48	0.2
MSVD	0.63	0.26	0.36	1	0.2
LSMDC	0.61	0.28	0.4	0.52	1

(b) Ratio of shared video clips per dataset

Table D.2: Number/Ratio of shared video clips in the datasets’ generated pairs P_{gen} .

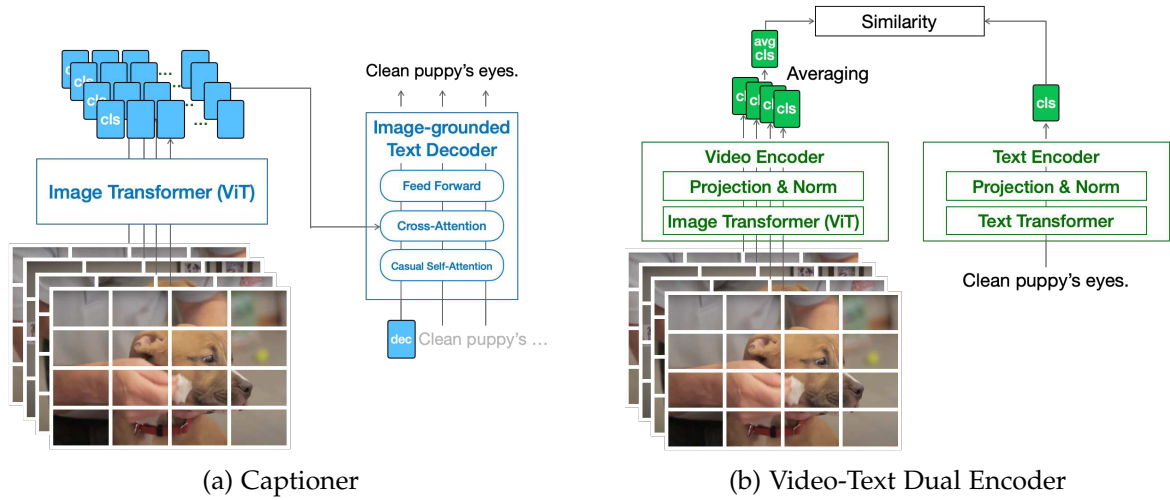


Figure D.3: Schematic visualization of (a) the video captioner architecture; and (b) the video-text dual encoder model.

Video-Text Dual Encoder. In the pseudo matching, filtering, and retrieval steps of our In-Style method, we use the video-text dual encoder model. We initialize the dual encoder model from image-text pretrained BLIP [LLXH22b] or CLIP [RKH⁺21b] models, which we extend to video-text models. Specifically, we obtain a video embedding by averaging image representations of m uniformly sampled frames, as shown in Figure D.3b. By default, eight frames are used $m = 8$ during training and $m = 12$ during evaluation. But m is increased to 64 during evaluation on the fine-grained DiDeMo dataset as in [LJZ⁺22, WLF⁺23]. To obtain image representation, we feed image patches into ViT transformer [DBK⁺21b], and the “cls” output token is later projected by a linear projection into common embedding space and further normalized. The text representation is obtained by projecting and normalizing an output “cls” token in the case of BLIP and an output “eot” token in the case of CLIP.

D.3 IMPLEMENTATION DETAILS

EOA Model Details. In case of the EOA [SCR⁺22] architecture, we use the model variant with S3D visual backbone with a frozen S3D backbone pretrained by Miech et al. [MAS⁺20] in the visual branch and a GoogleNews pretrained Word2vec model [MCCD13] in the text branch. These backbones are fixed and not trained. As initialization, we use EOA [SCR⁺22] weights pretrained with audio modality, but we discard the audio branch to consider text-video-only retrieval.

EOA Training Details. We follow [SCR⁺22] to train the EOA model. We compute one S3D feature per second [MAS⁺20] without any data augmentation; namely, we compute one feature per 16 frames, sampling frames with 16 fps and 224x224 resolution. We train with Adam [KB15a] optimizer, no weight decay, and a batch size of 128. We use a temperature of 0.05 in the loss function.

BLIP Evaluation Details. We would like to highlight that while evaluating a model with a BLIP backbone, we use dual encoder architecture (unimodal encoders), not a cross-attention architecture (named an image-grounded text encoder in [LLXH22b]), which performance on MSR-VTT dataset [XMYR16] was reported in the original paper [LLXH22b]. Dual encoder

models independently encode videos and texts into common embedding space, allowing for fast retrieval among thousands of videos by pre-computing video embeddings and calculating the similarity between text and video embeddings with a dot product (cosine similarity) [MAL⁺21]. Cross-attention architectures compute similarity by propagating video and text together in the model with cross-attention layers, attending all words and all spatio-temporal video patches to each other. Cross-attention architecture significantly boosts retrieval performance compared to dual encoder models [MAL⁺21]; however, it demands enormous computational overhead in the inference phase, requiring propagating all videos paired with a given text query to compute similarity. In the original paper [LLXH22b], BLIP performance on the MSR-VTT dataset was reported with the cross-attention model used to rerank 128 closest videos found by the dual encoder model. Since the majority of the methods [RKH⁺21b, SCR⁺22, PHA⁺21, MZA⁺19, LJZ⁺22, AYQ⁺21] leverage dual encoder architecture for video retrieval due to computational benefits, for comparison purpose we also base our method on dual encoder models.

D.4 QUALITATIVE RESULTS

Pseudo pairs P_{ps} and generated pairs P_{gen} . We provide additional qualitative results for pseudo pairs P_{ps} and generated pairs P_{gen} with the MSR-VTT dataset in Figure D.4 and the YouCook2 in Figure D.5. We observe on various datasets that generated captions capture content better than in initially obtained pseudo pairs after the matching step. It confirms our discussion of Table 6.5 (in the main part of the thesis), that generated pairs P_{gen} on average provide better improvement than pseudo pairs P_{ps} .

Text-video retrieval. We also demonstrate qualitative results of text-video retrieval on the MSR-VTT (Figure D.6) and YouCook2 (Figure D.7) datasets. We found that the proposed In-Style model retrieves more semantically similar videos to a given query compared to the zero-shot BLIP model.

D.5 LIMITATIONS

In this work, we rely on the pretrained large image-language models such as CLIP [RKH⁺21b] and BLIP [LLXH22b]. We consider this as an advantage and disadvantage at the same time. On one side, we show how to adapt such models to the input style of text queries, whereas, on the other side, we inherit all the biases that such models include [BHA⁺21]. Moreover, CLIP has the property that it can read text from the images [MTB22]; therefore, matching or filtering steps could suffer from that because some unrelated text (e.g., advertisement) appears on the frames (e.g., see Figure D.4).

While our motivation is to avoid annotation costs for aligning text-video pairs, we still rely on a video collection; namely, we utilize a large-scale dataset of YouTube videos. However, we note that such videos are easy to collect as they are available on YouTube and are not preprocessed; therefore, these videos include various types of potential noise from the web, such as advertisements, camera motion, low-quality videos, and others. Moreover, even though the distribution of our input text queries is not the same as the distribution of web support videos, we empirically observe that there is always some overlap between the distributions. Hence, we do not assume that our In-Style method will be helpful when input queries and test sets are from absolutely different domains like medicine (input text queries) and wildlife (test). As we show in our experiments, we obtain the most gain when the input text queries and test sets are from the same distribution.

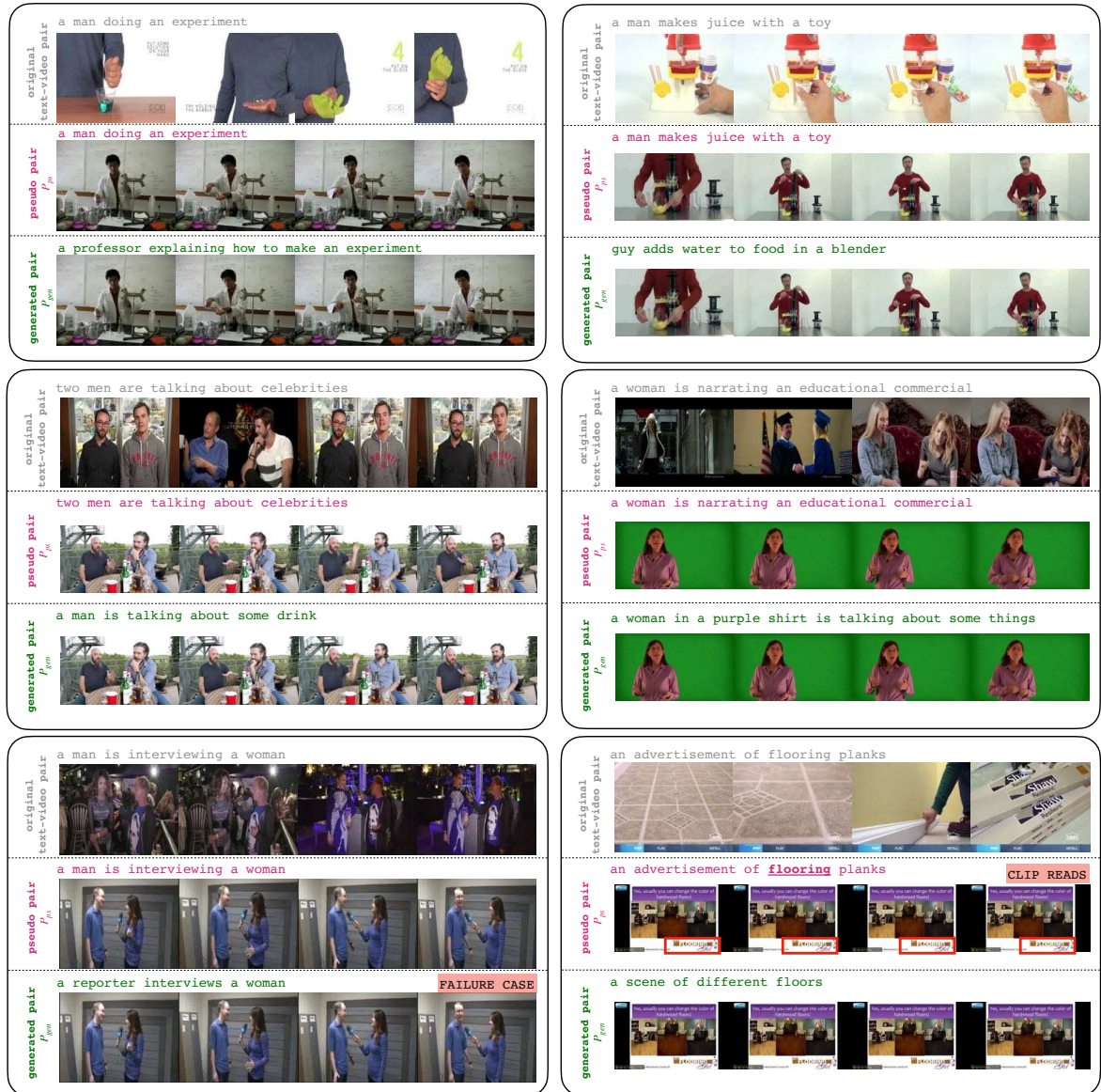


Figure D.4: Qualitative evaluation of P_{ps} and P_{gen} on the MSR-VTT. First, a text query is matched with one of the videos (a pseudo pair P_{ps}), and then, after the style transfer step, for each video, a new caption is generated in the same style but with updated content (a generated pair P_{gen})



Figure D.5: Qualitative evaluation of P_{ps} and P_{gen} on the YouCook2. First, a text query is matched with one of the videos (a pseudo pair P_{ps}), and then, after the style transfer step, for each video, a new caption is generated in the same style but with updated content (a generated pair P_{gen})

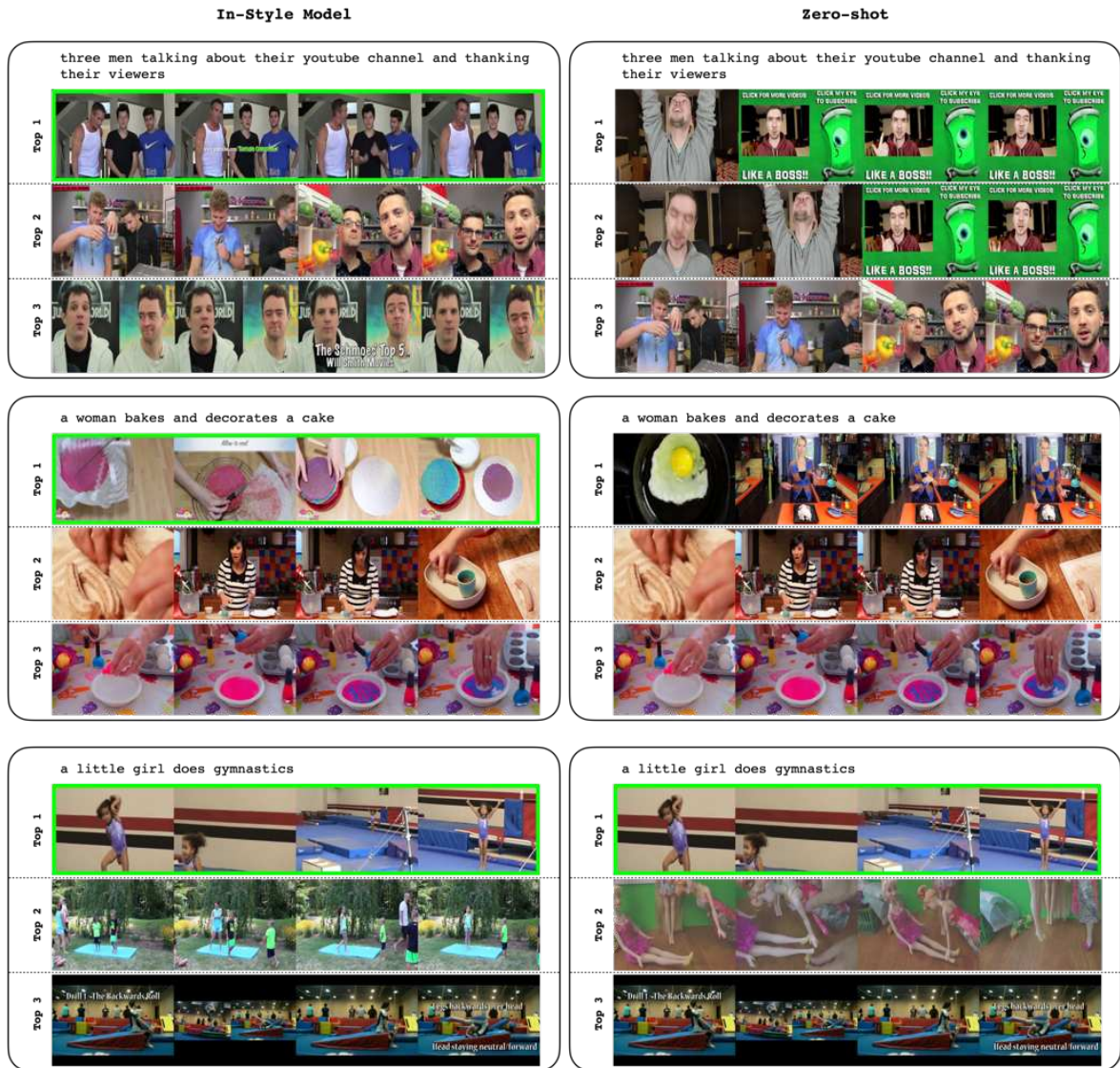


Figure D.6: **Qualitative evaluation of text-video retrieval on the MSR-VTT.** Retrieval examples for the proposed In-Style Model and zero-shot BLIP model. Each box shows the top-3 retrieved videos for a given text query. The correct video is highlighted with a green color.

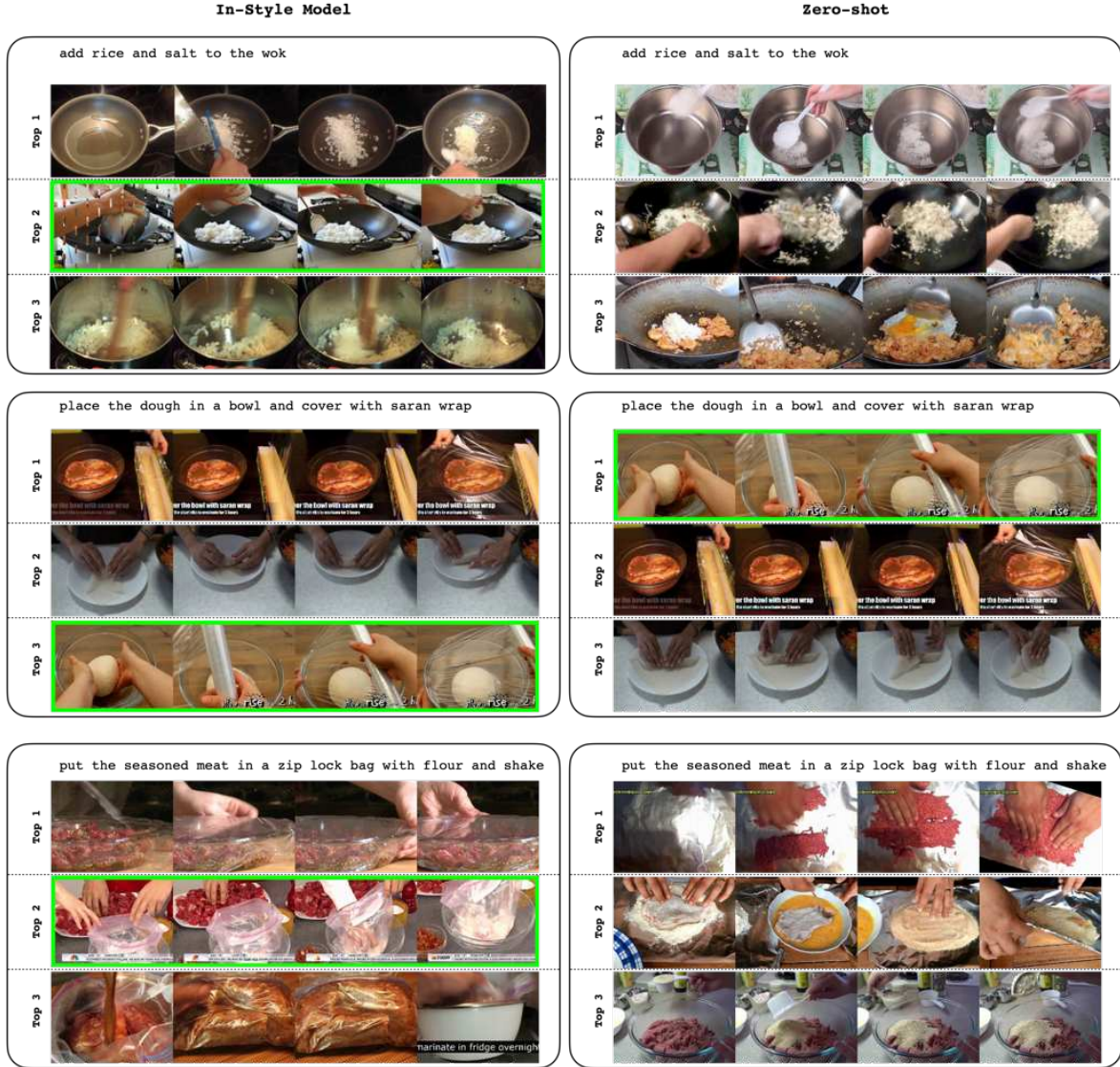


Figure D.7: **Qualitative evaluation of text-video retrieval on the YouCook2.** Retrieval examples for the proposed In-Style Model and zero-shot BLIP model. Each box shows the top-3 retrieved videos for a given text query. The correct video is highlighted with a green color.

APPENDIX – HOWTOCAPTION: PROMPTING LLMs TO TRANSFORM VIDEO ANNOTATIONS AT SCALE

This appendix contains supplementary material related to Chapter 7, *HowToCaption: Prompting LLMs to Transform Video Annotations at Scale*, of this thesis. We first discuss the experimental results of using MiniGPT-4 [ZCS⁺23] to generate visually grounded captions in Section E.1. Then, we provide additional experimental results in Section E.2. Then, we provide additional implementation details in Section E.3, statistics of the HowToCaption dataset in Section E.4, and qualitative examples in Section E.5. Finally, we discuss our method limitations in Section E.6 and responsibility to human subjects in Section E.7.

Contents

E.1	Grounding Captions to Video Content with MiniGPT-4	243
E.2	Additional Experimental Evaluation	244
E.2.1	Dataset Quality	244
E.2.2	Additional Comparisons with SOTA in Text-Video Retrieval	245
E.2.3	Additional Results in Prompt Engineering	246
E.2.4	Ablations of Filtering & Alignment Post-processing	246
E.2.5	Robustness Analysis	247
E.3	Additional Implementation Details	247
E.4	HowToCaption Dataset Statistics	249
E.5	Qualitative Examples	250
E.6	Limitations	251
E.7	Responsibility to Human Subjects	251

E.1 GROUNDING CAPTIONS TO VIDEO CONTENT WITH MINIGPT-4

Our generated captions with Vicuna-13B are based solely on ASR subtitles. To additionally ground the produced captions on visual content, we experiment with the recent MiniGPT-4 model [ZCS⁺23]. The MiniGPT-4 consists of the frozen Vicuna-13B model and a visual encoder with a Q-Former [LLXH22b] that projects visual features from an image into tokens in a language model embedding space that are later treated as word tokens in the Vicuna-13B model. To ground generated captions on the visual modality, we create a grid image from 4 uniformly sampled frames from a video clip and slightly adapt the prompt to encourage the LLM to utilize the provided image for generating captions (Table E.1). We apply our approach to obtain visually grounded captions with the MiniGPT-4 model and obtain *HowToCaption-grounded*. For this dataset, we follow exactly the same hyperparameters that we use for HowToCaption. In Table E.2, we evaluate the downstream retrieval performance of the T-V model trained on HowToCaption-grounded. The dataset shows mixed results compared to the HowToCaption; while it is beneficial for the MSR-VTT and the MSVD dataset, performance on the YouCook2 dataset drops. To facilitate further analysis, we will release both caption sets: the ASR-based only HowToCaption, produced by the Vicuna-13B, and HowToCaption-grounded,

Vicuna-13B	MiniGPT-4
I will give you an automatically recognized speech with timestamps from a video segment that is cut from a long video. Write a summary for this video segment. Write only short sentences. Describe only one action per sentence. Keep only actions that happen in the present time. Begin each sentence with an estimated timestamp. Here is this automatically recognized speech: <ASR with timestamps>	I will give you an automatically recognized speech with timestamps and an image with four frames from a video segment that is cut from a long video. Write a summary for this video segment based on both: video frames and speech . Write only short sentences. Describe only one action per sentence. Keep only actions that happen in the present time. Begin each sentence with an estimated timestamp. Here is the image with four frames: <grid-image here> . Here is the automatically recognized speech: <ASR with timestamps>

Table E.1: **Prompts for the Vicuna-13B and MiniGPT-4 models.** Difference is highlighted with bold.

Dataset	YouCook2				MSR-VTT				MSVD				LSMDC			
	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR
HowToCaption (Vicuna-13B)	13.4	33.1	44.1	15	37.6	62.0	73.3	3	44.5	73.3	82.1	2	17.3	31.7	38.6	29
HowToCaption-grounded (MiniGPT-4)	12.4	29.8	39.9	20.5	38.3	62.5	73.2	2	46.2	73.9	82.5	2	16.8	31.0	38.7	27

Table E.2: **Comparison of HowToCaption and HowToCaption-grounded datasets obtained with Vicuna-13b and MiniGPT-4 large language models, respectively.** For each dataset, we train a T-V model and report downstream zero-shot text-video retrieval performance.

produced by the MiniGPT-4.

E.2 ADDITIONAL EXPERIMENTAL EVALUATION

E.2.1 Dataset Quality

In this section, we provide an additional quantitative evaluation of the quality of captions in our HowToCaption dataset.

Evaluation Using the HIREST Dataset. First, we evaluate the caption quality using the HIREST dataset [ZCK⁺23] as the ground truth. The HIREST dataset contains step captions for a small subset of video clips from the HowTo100M dataset. Since the video clip boundaries of the HIREST dataset, our HowToCaption dataset, and the HowTo100M dataset differ, we use a subset of 423 HIREST (video clip, caption) pairs that have corresponding (video clip’, caption’) and (video clip”, caption”) pairs in the HowToCaption and HowTo100M datasets respectively, such as the temporal boundaries of (video clip and video clip’) overlaps with > 0.5 IoU, as well as (video clip and video clip”) overlaps with > 0.5 IoU. Using HIREST captions as ground truth, we report the quality of captions in the HowTo100M and HowToCaption datasets using standard metrics such as BLEU@4 and METEOR, as shown in Table E.3. Despite the common disadvantages of classical metrics, which focus more on wording rather than semantics, we observe consistent improvement in the captions of our HowToCaption dataset over the ASR subtitles.

	BLEU@4 METEOR ROUGE-L CIDEr			
HowTo100M with ASRs	0.4	9.8	11.4	3.1
HowToCaption (ours)	1.0	9.3	13.6	29.6

Table E.3: **Caption quality evaluation in the HowTo100M and HowToCaption datasets using the HIREST dataset as the ground truth.** We use a subset of 423 HIREST video clip-caption pairs that temporally overlap with video clip-caption pairs in both the HowToCaption and HowTo100M datasets with > 0.5 IoU. We report standard text generation metrics.

Prompt	YouCook2		MSR-VTT		MSVD		LSMDC		Average	
	R10 $\uparrow$	MR $\downarrow$	R10 $\uparrow$	MR $\downarrow$	R10 $\uparrow$	MR $\downarrow$	R10 $\uparrow$	MR $\downarrow$	R10 $\uparrow$	MR $\downarrow$
I will give you an automatically recognized speech with timestamps from a video segment that is cut from a long video. Write a summary for this video segment. Write only short sentences. Describe only one action per sentence. Keep only actions that happen in the present time. Begin each sentence with an estimated timestamp. Here is this automatically recognized speech: <ASR with timestamps in the format "n"s: "ASR"> (ours)	40.6	19	72.0	3	81.6	2	37.7	30	58.0	13.5
Modification: Write a summary for this video segment. → Write a likely summary for this video segment.	40.8	18.5	71.4	3	81.5	2	37.7	30	57.9	13.4
Modification: Write a summary for this video segment. → Write a creative summary for this video segment.	40.0	19	71.6	3	81.2	2	37.8	27	57.7	12.8
Modification: <ASR with timestamps in the format "n"s: "ASR"> → <ASR with timestamps in the format "minutes": "seconds": "ASR subtitle">	40.8	18.5	71.5	3	81.2	2	37.2	29	57.7	13.1

Table E.4: **Additional experiments with LLM prompts.** We report modifications that we have done in comparison to our default prompt, which is **highlighted**. With each prompt, we obtain 2M video-text pairs from 100k HowTo100M videos that we later use for T-V model training (low-recourse setup). Downstream zero-shot text-video retrieval performance is reported.

User Study. To quantitatively evaluate the “human-written”-like quality of our captions, we performed a user study with 60 random captions: 20 from HowToCaption, 20 ASR subtitles, and 20 captions from the downstream datasets, namely YouCook2, MSR-VTT, MSVD, and LSMDC (5 captions per dataset), which served as a control group. We asked users to identify *human-written video captions*. Using majority voting from 30 participants, 90% of the control group, 50% of HowToCaption, and only 15% of ASR captions were chosen as “human-written” captions. This demonstrates that our captions are significantly more “human-written” than the ASR subtitles.

E.2.2 Additional Comparisons with SOTA in Text-Video Retrieval

In addition to comparing dual encoder-only models in zero-shot text-to-video retrieval, in Table E.5, we compare methods that use a visual-text matching head for *re-ranking* the best candidates [LXT⁺22, CLW⁺24]. Specifically, we evaluate our model with all BLIP architecture blocks used in video captioning evaluation (as described in Section 7.4.4 of the main part of the thesis and in Appendix E.3) by re-ranking 128 best candidates from dual encoder model

Method	V. Encoder	YouCook2				MSR-VTT				MSVD				LSMDC			
		R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR
BLIP* [LLXH22b]	ViT-B	10.7	24.1	32.3	44.5	41.4	63.3	72.8	2	45.6	71.3	79.7	2	20.5	36.5	44.2	18.5
BLIP*, HowToCaption-finetuned	ViT-B	18.15	38.68	50.4	10	44.3	66.6	76.6	2	49	76.2	84.2	2	19.6	34.3	42.9	18
BLIP, COCO-finetuned [LLXH22b]	ViT-B	-	-	-	-	43.3	65.6	74.7	2	48.7	75.8	83.1	2	22.3	38.4	44.1	18
Unmasked Teacher-17M [LWL ⁺ 23]	ViT-L	-	-	-	-	42.6	64.4	73.1	-	49.9	77.7	85.3	-	25.2	43	50.5	-
mPLUG, COCO-finetuned [LXT ⁺ 22]	ViT-L	-	-	-	-	44.3	66.4	75.4	-	-	-	-	-	-	-	-	-
mPLUG-2 [XY ⁺ 23]	ViT-L	-	-	-	-	47.1	69.7	79	-	-	-	-	-	24.1	43.8	52	-
VAST [CLW ⁺ 24]	ViT-G	-	-	-	-	49.3	68.3	73.9	2	-	-	-	-	-	-	-	-
VAST ⁺ [CLW ⁺ 24]	ViT-G	15.7	35.2	45.6	14	48.5	71.2	79.8	2	50.6	76.2	84.1	1	23.2	40.9	48.9	12
VAST ⁺ , HowToCaption-finetuned	ViT-G	19.7	43.6	53.9	8	50	73.2	81.4	1	54.8	80.9	87.2	1	27.7	46.5	54.6	7

Table E.5: **Comparison of zero-shot text-to-video retrieval methods with re-ranking.** We include methods that perform a re-ranking of the top-N retrieved candidates using a visual-text matching head. *BLIP_{CapFilt-L} (initialization of our model). ⁺Text-video model only (without audio and subtitles).

predictions using a visual-text matching head. Moreover, we finetune the state-of-the-art text-video VAST model [CLW⁺24] for 9k interactions on the HowToCaption dataset with a batch size of 64, a learning rate of 5e-06, using 4 frames per video clip, and following other training/evaluation parameters of VAST. We observe that finetuning on the HowToCaption dataset boosts the performance of both models. Furthermore, the text-video VAST model finetuned on the HowToCaption dataset outperforms all other models.

E.2.3 Additional Results in Prompt Engineering

In Table E.6, we provide an additional evaluation of language prompts. First, we experiment with phrases such as “write a *likely* summary...” and “write a *creative* summary...”. While the keyword “likely” almost does not change downstream performance, the keyword “creative” is not beneficial for 3 out of 4 datasets. We also experiment with another timestamp format in the LLM prompt. Namely, instead of using “n”s (such as 0s, 65s), we use “minutes”：“seconds” format (such as 00:00, 01:05). We found that simple timestamp format “n”s results in a higher performance.

E.2.4 Ablations of Filtering & Alignment Post-processing

We present ablations of our alignment & filtering post-processing in Table E.7. In Table E.7a, we ablate two modifications of the finetuning and alignment processes for the second round of filtering & alignment. We observe that the dataset obtained after the second round of filtering & alignment without these modifications shows lower performance than the dataset obtained with the first round (using the BLIP model). We attribute this to forgetting during finetuning. However, we note that both proposed modifications boost performance, as well as their combination.

In Table E.7b, we also analyze if more than two rounds of filtering & alignment lead to a better quality dataset. We employ 20k iterations of finetuning of the T-V model on the obtained dataset after each filtering & alignment round. We do not observe any performance boost with more filtering & alignment rounds.

We further ablate the range of alignment time offsets T and filtering threshold κ used in our alignment & filtering method in Table E.7c and Table E.7d respectively. We found that

Prompt	YouCook2		MSR-VTT		MSVD		LSMDC		Average	
	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓
I will give you an automatically recognized speech with timestamps from a video segment that is cut from a long video. Write a summary for this video segment. Write only short sentences. Describe only one action per sentence. Keep only actions that happen in the present time. Begin each sentence with an estimated timestamp. Here is this automatically recognized speech: <ASR with timestamps in the format "n"s: "ASR"> (ours)	40.6	19	72.0	3	81.6	2	37.7	30	58.0	13.5
Modification: Write a summary for this video segment. → Write a likely summary for this video segment.	40.8	18.5	71.4	3	81.5	2	37.7	30	57.9	13.4
Modification: Write a summary for this video segment. → Write a creative summary for this video segment.	40.0	19	71.6	3	81.2	2	37.8	27	57.7	12.8
Modification: <ASR with timestamps in the format "n"s: "ASR"> → <ASR with timestamps in the format "minutes": "seconds": "ASR subtitle">	40.8	18.5	71.5	3	81.2	2	37.2	29	57.7	13.1

Table E.6: **Additional experiments with LLM prompts.** We report modifications that we have done in comparison to our default prompt, which is **highlighted**. With each prompt, we obtain 2M video-text pairs from 100k HowTo100M videos that we later use for T-V model training (low-recourse setup). Downstream zero-shot text-video retrieval performance is reported.

alignment with up to $T = 10$ seconds offsets and filtering with the threshold that selects 25M most similar video-text pairs result in the highest performance.

E.2.5 Robustness Analysis

We further analyze the robustness of our HowToCaption method to noisy input. First, we examine how the method performs if the video channel is corrupted, specifically containing only empty black frames. We replace all 1.2M videos with videos of the same length but containing only black frames while keeping the original ASR subtitles of the videos. Consequently, the generated captions from subtitles remain the same, but the videos are corrupted. After our alignment & filtering procedure, we find that only $\sim 1.8\%$ of generated captions are matched. Therefore, our method filters out almost all captions for corrupted videos. Furthermore, we test the case of data where ASR subtitles do not match the videos. For each video among the 1.2M videos, we randomly assign ASR subtitles from another video of the dataset. In this case, generated captions from subtitles are the same, but they do not correspond to the videos. We find that after applying the HowToCaption method to such input data, only $\sim 2.4\%$ of captions are matched after alignment & filtering. Therefore, almost all noisy and irrelevant captions are filtered, indicating our method’s robustness for noisy data.

E.3 ADDITIONAL IMPLEMENTATION DETAILS

For our T-V model, we follow BLIP’s [LLXH22b] dual encoder architecture with a ViT-B/16 visual encoder and a BERT_{base} textual encoder, which are initialized with BLIP_{CapFilt-L} pretrained weights. Following BLIP [LLXH22b], we also use an extension of the loss (Equation 1 in the

Caption post-processing	YouCook2		MSR-VTT		MSVD		LSMDC		Average	
	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓
Alignment & filtering (using the BLIP)	42.4	17	71.7	3	82.2	2	38.5	29.5	58.7	12.9
Alignment & filtering after second round + regularization L_{align} + averaging similarities of the finetuned and original model + regularization L_{align} + averaging similarities of the finetuned and original model (ours)	42.4	17	69.4	3	81.2	2	38.1	33	57.8	13.8
	44.3	15	71.9	3	81.9	2	39.0	28	59.3	12.0
	43.7	15	72.8	3	82.0	2	39.6	27	59.5	11.8
	44.1	15	73.3	3	82.1	2	38.6	29	59.5	12.3

(a) Components of our filtering & alignment method.

Caption post-processing	YouCook2		MSR-VTT		MSVD		LSMDC		Average	
	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓
Alignment & filtering (using the BLIP) = 1 round	42.4	17	71.7	3	82.2	2	38.5	29.5	58.7	12.9
Alignment & filtering after 2'nd round (ours)	44.1	15	73.3	3	82.1	2	38.6	29	59.5	12.3
Alignment & filtering after 4'th round	44.5	15	72.2	3	81.8	2	38.6	29	59.3	12.3

(b) Rounds of alignment & filtering with finetuning of the T-V model after each round.

Alignment time offsets $\delta \in \mathbb{Z}, \delta \leq T$	YouCook2				MSR-VTT				MSVD				LSMDC				Average			
	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR
$T = 0$ seconds (no align.)	12.9	32.1	43.5	15	38	61.5	71.9	3	44.1	72.9	81.9	2	16.8	31.0	37.8	30.5	28.0	49.4	58.8	12.6
$T = 10$ seconds	13.4	33.1	44.1	15	37.6	62.0	73.3	3	44.5	73.3	82.1	2	17.3	31.7	38.6	29	28.2	50.0	59.5	12.3
$T = 20$ seconds	13.2	32.0	42.9	16	37.9	62.4	72.8	3	44.8	73.4	82.2	2	16.7	31.9	39.0	28	28.2	49.9	59.2	12.3

(c) Range of time offsets during alignment.

Number of pairs after filtering (varying κ)	YouCook2				MSR-VTT				MSVD				LSMDC				Average			
	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR
15M pairs	13.5	32.2	43.0	16.5	37.6	62.2	72.6	3	45.1	73.4	82.2	2	16.7	32.3	38.5	28.5	28.2	50.0	59.1	12.5
25M pairs	13.4	33.1	44.1	15	37.6	62.0	73.3	3	44.5	73.3	82.1	2	17.3	31.7	38.6	29	28.2	50.0	59.5	12.3
40M pairs	12.9	32.7	43.7	15	37.8	60.5	71.3	3	44.1	73.3	81.8	2	16.7	31.9	37.8	29.5	27.9	49.6	58.7	12.4

(d) Filtering threshold κ .

Table E.7: **Ablation of our alignment & filtering method.** With each post-processing variant, we obtain a dataset that we later use for T-V model training. Downstream zero-shot text-video retrieval performance is reported. Options used to obtain the main results are highlighted.

paper) with soft labels produced by a momentum encoder and a memory bank that keeps additional text and video embeddings from the previous iterations. We train the model for 300k iterations using AdamW [LH19] with a batch size of 128, a learning rate of $1e-6$, and a weight decay of 0.05. We use a memory bank of 2048 and smooth labels with a parameter of 0.6. Training augmentation is cropping with a scale $[0.5, 1]$. For model finetuning in the alignment & filtering step, we use 20k training iterations and regularization parameter $\alpha = 0.1$.

Video Captioning Details. For our video captioning experiments, we finetune the BLIP [LLXH22b] model with all architecture blocks, including the image encoder, the text encoder, the image-grounded text encoder, and the image-grounded text decoder. Following [LLXH22b], we use all three BLIP losses: image-text contrastive loss, image-text matching loss, and language modeling loss for finetuning. If not stated otherwise, we use the same hyperparameters as in our main text-video retrieval experiments, including the ViT-B/16 visual encoder and the BERT_{base} textual encoder initialized with the BLIP_{CapFilt-L} weights, as well as the same learning rate, batch size, etc.

We finetune the model for 200k iterations on the HowToCaption dataset. Then, we finetune the image captioning model: the image encoder and the image-grounded text decoder (following [LLXH22b]) on the corresponding training set of the YouCook2, MSRVT, or MSVD datasets. Note that the MSRVT dataset has different splits for retrieval and captioning evaluation. Following standard practice [LLL⁺22, TWL⁺21, SNAS22], we use 6.5k videos for training and 3.5k for testing of video captioning. For finetuning on downstream datasets, we use a batch size of 16, a learning rate of $1e-5$, and a learning rate scheduler with cosine decay. We sample 16 frames for training and evaluation. We finetune for 10 epochs on the YouCook2, 30 epochs on the MSRVT, and 40 epochs on the MSVD. For captioning performance evaluation, we set a number of beams = 1 and a maximum length = 20, as in [LLL⁺22].

Text-to-Video+Audio Retrieval Details. For text-to-video+audio experiments, we train a multimodal Everything-At-Once (EAO) model [SCR⁺22] with frozen S3D features [MAS⁺20] on our HowToCaption dataset. Since our dataset addresses the issues of high correlation between audio and text, we exclude additional regularizations that aim to prevent the model from learning shortcuts, i.e., simplifying the task to speech recognition from audio while ignoring the video. by simply performing speech recognition in audio and ignoring video. Specifically, we omit shifting audio timestamps with respect to video clip boundaries and assigning lower weights to the audio loss. All hyperparameters are kept the same as in EAO[SCR⁺22].

E.4 HOWTOCAPTION DATASET STATISTICS

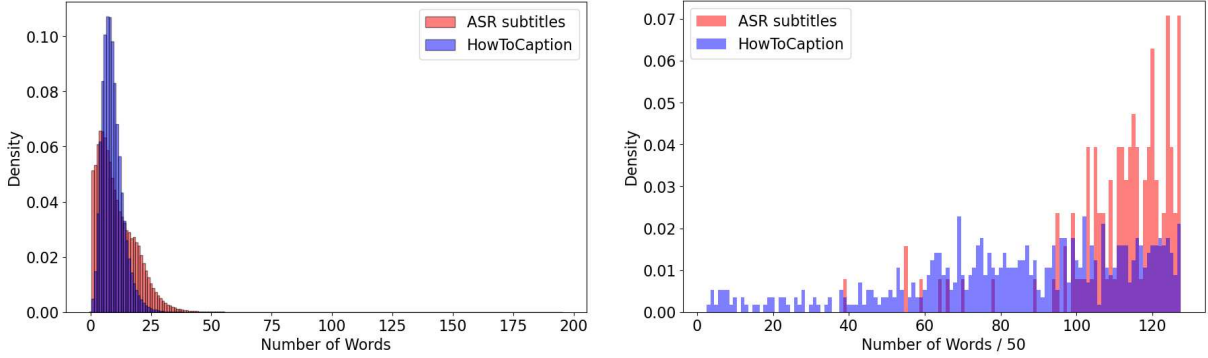
In this section, we present the statistics of our HowToCaption dataset. Our goal is to demonstrate the scale and diversity of the captions in the proposed dataset.

Caption Length. To better understand the scale of our dataset, we compute caption length statistics. We analyze captions both at the video clip level and at the video level (when combining captions from all clips belonging to the same video). We randomly sample 5000 videos from HowToCaption and use a spaCy tokenizer [HMLV⁺20] to count words. The resulting histograms of caption length are shown in Figure E.1 and statistics in Table E.8. On a sentence level, our dataset has shorter captions on average (9.03 words) compared to the original ASR subtitles (10.97 words). Our captions also have a smaller standard deviation (4.36 vs. 7.91), indicating a more consistent length distribution.

Language Diversity. We also compare the language diversity of the ASR subtitles and our

Dataset	Standard statistics			Diversity↑ %diverse verbs	n -grams diversity↑			Verb n -grams diversity↑		
	$ V $	#word /caption	#word /video		1-gram	2-gram	3-gram	1-gram	2-gram	3-gram
ASR subtitles	45905	10.96	909.34	77.09	1.01	17.57	50.24	1.13	19.88	61.64
HowToCaption	36204	9.03	581.27	82.55	1.25	21.36	53.95	1.01	25.9	76.65

Table E.8: **Language statistics.** $|V|$ is the vocabulary size. #word/caption is the number of words per caption. #word/video is the number of words per all captions in a video. %diverse verb is the percentage of diverse verbs. All numbers are obtained from 5000 randomly sampled videos.



Histogram of caption lengths.

Histogram of total length of all captions for a video. The scale of the x-axis is divided by 50.

Figure E.1: **Caption length statistics of our HowToCaption dataset.** We randomly sample 5000 videos to plot the distributions.

captions. We measure language diversity from two perspectives: 1) diversity based on the presence of distinct words or verbs and 2) diversity of word/verb n -grams across the captions, providing insights into the varied combinations of words/verbs used in our captions. In our analysis (Table E.8), we follow [GTCWP20] and calculate the percentage of diverse verbs (that are not in the top 5 most frequent verbs) relative to all verbs. Following [SRAH⁺17], we also compute the unique-to-total ratio for word unigrams, bigrams, and trigrams (e.g., the ratio between the number of unique word unigrams to the total number of word unigrams over all captions). We further use the spaCy toolkit [HMLV⁺20] to extract and lemmatize verbs and calculate the unique-to-total ratio for *verb* unigrams, bigrams, and trigrams. The results in Table E.8 show that the captions in the HowToCaption dataset have higher language diversity than ASR subtitles across almost all measures except on verb unigram. We observe that the longer action sequences in HowToCaption are more diverse than ASR subtitles, which demonstrates the high quality of our dataset.

E.5 QUALITATIVE EXAMPLES

HowToCaption Dataset. We present an extension of Figure 7.3 from the main part of the thesis with video-text examples of our HowToCaption dataset and corresponding ASR subtitles in Figure E.2. We see that our HowToCaption method effectively transforms noisy ASR subtitles into proper captions, leveraging the complete ASR context for caption generation. We demonstrate additional video-text examples of our HowToCaption dataset in Figure E.3 and

Figure E.4. In Figure E.4, we also showcase instances of failure cases. One such case involves a failure where the LLM was unable to generate a caption and instead copied the input ASR subtitles: “DP Move Safe lets operators get out of the classroom...” However, in this example, the ASR subtitles contain a third-person video description with a subject+verb+object sentence structure that justifies the copying input description without modification. Other failure cases include video-caption pairs, where the caption corresponds to the video only partially, e.g., “Cover it with lid” action is not visible on the video while “until the seviayan is cooked” is visible.

LLM Caption Generation. In Table E.9, we showcase captions generated by the Vicuna-13B model, presenting both the input ASR subtitles and their corresponding generated captions for comparison. We observe the LLM is able to transform scrambled ASR subtitles into “human-written-like” descriptions. However, we also note that sometimes LLM fails to produce descriptions. We present some failure cases in Table E.10, which include 1) direct input repetition: instances where the LLM duplicates ASR input without modification; 2) ineffective reformulation: the LLM attempts to convert ASR content into descriptions using ineffective structures like “A person says...”; 3) failure to follow the requested structure: instances where the LLM output doesn’t follow “a timestamp: a sentence” structure for output, e.g., using “Summary: ” to write a video description without timestamps.

E.6 LIMITATIONS

Our method relies on pretrained large foundational models, including the large language model Vicuna [CLL⁺23] and the vision-language model BLIP [LLXH22b]. Consequently, our HowToCaption method and the proposed HowToCaption dataset may inherit limitations present in these models. Notably, large language models have several shortcomings, such as biases from their training data, which can lead to the generation of potentially misleading content and the propagation of societal biases [Ray23]. Additionally, since our text-video model is initialized from the BLIP model that was pretrained on curated and filtered datasets, it might be less robust to noisy low-quality videos in our alignment & filtering step. In our robustness analysis, we found that the model is capable of filtering noisy input, but 1-2% of noisy data is still passing the filter. Finally, since our dataset is sourced from the HowTo100M [MZA⁺19] dataset, it follows the same data distribution, focusing solely on “how-to” topics. While we show improvement across many different tasks, it may limit its applicability to certain downstream tasks.

E.7 RESPONSIBILITY TO HUMAN SUBJECTS

Our HowToCaption dataset is sourced from the publicly accessible HowTo100M dataset [MZA⁺19], which was collected from YouTube. The video content consists of user uploads and is publicly available. However, since the dataset provides only video IDs for download, users who opt out of YouTube are consequently excluded from the HowTo100M and HowToCaption datasets. We are not aware whether consent was obtained from the users to be included in the original dataset. The dataset may include celebrities or other YouTube-famous individuals.



Caption (118s-126s): Matt Swanson gives a tip to use buckets to direct the path of the ball

ASR: 7s: hi i'm matt swanson
 9s: with matt swanson's we'll go is gonna help me change the direction of your ball play
 16s: if you're struggling with a slice or a hook and it's moving too much in that direction
 21s: i'm gonna give you a little tip that we use
 25s: that's very easy that you can do when you're out to help
 84s: change that
 91s: you're slicing now
 109s: too much
 116s: so use these buckets when you're out at the range
120s: move them around to help direct the path
 123s: make sure the clubface is closing if you're trying to get rid of the slice opening
 128s: if you're trying to hit a fade use these tips and you'll get better



Caption (187s-195s): Making a bow with two colors

ASR: 185s: so if you don't get your your ribbon twisted there's no up or down side to it
191s: so it's not going to really show
 193s: now once i get my three loops on my lighter color i'm going to make a little loop and this is basically just to hide my wire
 202s: that i'm going to use
 205s: the wire i use either a teen or a 20 gauge wire and what i'm going to do
 210s: this little loop is going to be my hide for the wire
 215s: so i just slide that through like that and pull real tight and twist
 227s: that will keep your loops good and snug once you get done
 237s: just kind of work your your loops around
 243s: and now you have a bow that's made with two colors
 246s: these are great for easter baskets you can use those for mother's day
 251s: we have an assortment of colors so we even have some for the holidays for fall
 257s: it's the type that you can use any time of the year and it makes great bows



Caption (87s-95s): Dog wants to hang out near dirt or other dogs with bones to acquire more bones

ASR: 87s: so this is stage one of hiding the bone
 90s: burying the bone
 90s: there's so much more involved
 92s: let's watch how she behaves
 94s: when she comes back from burying the bone she becomes a bit annoying
 97s: she wants to hang out next to dirt or somebody else who has a bone so that she might acquire yet another bone
 103s: now what i have found in my experience with her and remember every dog is different
 107s: nobody does things the same way but she lets those bones ferment for about two days before she goes back and finally retrieves the bone
 115s: and then she sits down and enjoys it
 116s: and then something makes it really special
 ...



Caption (68s-76s): Make sure the bottle stays together

ASR: 0s: hi there the amateur scientist here
 4s: and today in this video i'm going to show you how to make a very cheap and from recycled materials
 12s: a mosquitoes trap
 14s: now because we all know that mosquitoes can bite and make you itch and they are very uncomfortable
 20s: so today in this video i'm going to show you how to make a mosquito trap from a plastic bottle and on their amulet or zoom
 28s: so let's get into the video
 30s: first of all we're going to need a plastic bottle and i and some very cheap one
 45s: you could also use these for sugar
 61s: so first grab the bottle and and put the upper part into the lower part
69s: but this yeah and it just stays or it won't get off
73s: it's busy here
 74s: good deeper cheap wine
 79s: hello this with it like this or just a little bit more like that
 ...

Figure E.2: Extended example of video-captions pairs from our HowToCaption dataset (an extension of Figure 7.3 of the main part of the thesis). The ASR subtitles within the corresponding video clip are bolded. We note that some details in the generated captions are derived from a long ASR context.



Caption: Video segment starts with a shot of David's face, which is described as funny

ASR: and the bottom is actually has holes in it because it gets so incredibly hot so you cannot submerge it in water so we ask you to just rinse it out real quick look at david's face he is so funny



Caption: Adds two cans of red kidney beans to the chili

ASR: you could also use a vegetable broth all right so we're mixing this well



Caption: Adding chopped onions and green chillies to the pan

ASR: once the oil is hot enough we will add our onions and green chillies we need to cook the onions for some time maybe like 2 to 3 minutes until you start noticing that the colors of the onion have changed



Caption: Shows where wire runs along inside of vehicle

ASR: then ran alongside the gasket right here and runs down here and then this we took off and then ran the wiring in through here put this back down



Caption: Brutus is encouraged to swallow his medication

ASR: if i put it in a piece of food he'll chew it up and spit it out he knows oh baby i've never seen him do this though



Caption: She explains the charm tool's little piece of metal acts as a spacer to hold the charms open

ASR: has this little piece of patootie metal right here that acts as a spacer to hold the charms open so that gap is visible in the back so it's easier to slip the charms on and off



Caption: Paints the top part

ASR: i also notice how the blue continues onto the front of him just like right there so be careful with that next you take the white color and you would paint the webbing that he's hanging from here and also his eyes



Caption: You are making a rose petal exfoliating face scrub

ASR: you guys one of our favorite diys ever had to do with rose petals so we thought let's make another one

Figure E.3: Examples video-captions pairs from our HowToCaption dataset. Since ASR subtitles' timestamps do not always correspond to the timestamps of video clips from the HowToCaption dataset, we show ASR subtitles that intersect with video clip boundaries.



Caption: Soft bed in the car

ASR: but honestly when people see a lone prius in the parking lot no one thinks hey i wonder if someone's sleeping in there because come on it's a prius what fits in my car i have a soft bed i have a closet blackout curtains a desk kitchen table and chair a pantry a bike a laundry basket travel kit for emergencies



Caption: Asks viewers to choose favorite pet

ASR: i think i like them all for different reasons so that's hard so they can all three be my favorite can't they and look at her little friend



Caption: DP Move Safe lets operators get out of the classroom and out of their truck faster where they learn how to perform every task and do it safely (failure)

ASR: dp move safe lets operators get out of the classroom and out of their truck faster where they learn how to perform every task and do it safely



Caption: Cover it with a lid for 15 minutes until the seviayan is cooked (failure)

ASR: today, we will prepare a sweet recipe.. ..called 'sevaiyan' (vermicelli) so come on, let's see how to make sweet seviayan.



Caption: Walk upstairs to show light in the ceiling

ASR: i placed them over here because it's a little bit lighter on this side of the stair case then the other side there's only one light in the ceiling here so i'm gonna walk upstairs and i'm gonna let you see it from the top of the stairs one more time



Caption: The speaker adds white paint to the brush to keep the color bright

ASR: so i lay it on with the flat of the brush which deposits it a little heavier it holds up a little better and notice i keep adding white as ...



Caption: Outdoor Edge has instructional gated processing DVDs available on their website (failure)

ASR: this is one of the big issues with large diameter sausage products remember processing your own wild game animal can be fun easy and very rewarding if you have the tools and the knowledge to do the job you're watching outdoor edges



Caption: The group harvests fresh berries from the farm (failure)

ASR: the kids and i are here at a local blue red patches to manage to harvest some fresh berries at pcc are fresh and frozen organic ...

Figure E.4: Examples video-captions pairs from our HowToCaption dataset. Failure cases are marked as (failure). Since ASR subtitles' timestamps do not always correspond to the timestamps of video clips from the HowToCaption dataset, we show ASR subtitles that intersect with video clip boundaries.

Input ASR subtitles	Generated captions
os: hi guys it is bill with septic flow 4s: i am here at a brand new construction that i'm actually working on 8s: here 9s: on the back is the septic field 10s: we've already had bulldozing contractor actually cover up the the septic system 15s: but i'm going to show you why 17s: you want to use septic flow to pre perk the system 22s: septic flow is a product that actually helps get rid of sodium and phosphate buildup and we're actually going to just dump it right here in this two - inch 29s: it goes right out there 29s: we're going to run some water behind it for new construction 33s: the reason you want to do that is because we are actually pre perking the system getting it ready to take those phosphates and sodium buildup 41s: that happens through all different kinds of things 43s: you know whether you're using 44s: i've had people ask me can you use organic detergent 47s: will that prevent sodium buildup 50s: absolutely not 50s: soap by nature of the saponification process that it goes through it's just part of it	os: Bill is at a new construction site. 4s: He is showing the septic field. 8s: The bulldozing contractor has covered up the septic system. 10s: Bill explains why they use septic flow. 17s: Septic flow helps get rid of sodium and phosphate buildup. 22s: Bill demonstrates how they will dump septic flow in a two-inch area. 29s: They will run water behind it for new construction. 33s: Bill explains that they are pre-perking the system. 41s: The system needs to be prepared to take away phosphates and sodium buildup. 44s: People ask if using organic detergents can prevent sodium buildup. 50s: The answer is no, soap is part of the saponification process and will cause buildup.
104s: the great thing about the smart chicken 107s: with it being air chilled 108s: it's going to pick up a lot of these flavors when we put the the ginger the garlic 111s: and the sage in here 112s: it's just going to combine well with the meat especially while it's cooking on the grill 116s: all those flavors are just going to pop once the sandwich is complete and once it's all done 121s: now that we have it all mixed i'm just going to separate that out into two patties 129s: um you don't have to pat it up too much 131s: it takes a lot to get it on the grill so just kind of separate it out into two balls 136s: even 136s: and that's our chicken 138s: it's ready for the grill ...	104s: The speaker prepares the chicken by air-chilling it and adding ginger, garlic, and sage. 108s: The chicken will pick up the flavors while cooking on the grill. 111s: The speaker mixes the chicken and its seasoning. 116s: The flavors will pop once the sandwich is complete. 121s: The speaker separates the chicken mixture into two patties. 129s: The patties are not patted too much, as they will be grilled soon. 131s: The chicken is ready for the grill. 138s: The grill should be hot, around 400 degrees, for grilling the chicken. 141s: The chicken will flatten itself out and get a nice sear on the bottom when grilled. 149s: The speaker recommends not touching the chicken for about four minutes so the bottom can set up. 157s: The chicken will fall apart if touched too soon.
9s: hello home bruce and welcome to my channel 15s: so i am down to my last bottle of strawberry wine 17s: i've got some in a glass so i thought it would be a fantastic idea to make some more 23s: so it is a bit early in the season to be getting strawberries 27s: it has been rather cold so i am actually using frozen strawberries so i hope you enjoyed the video and i'm gonna have a drink 42s: so for this recipe i am actually going to be using a beauty bucket ...	9s: Hello and welcome to the channel. 15s: The host is down to their last bottle of strawberry wine. 17s: They have some wine in a glass and decide to make more. 23s: The host mentions that it's early in the season for strawberries. 27s: The host is using frozen strawberries in the recipe. 42s: The host is using a beauty bucket for the recipe. 46s: The bucket is an upcycled mayo container. 52s: The host mentions that a wide-mouthed bucket is ...

Table E.9: **Examples of caption generation with the Vicuna-13B.** Input ASR subtitles and output generated captions by the Vicuna-13B (without any post-processing). For brevity, lengthy examples are shortened with the symbol "...".

Input ASR subtitles	LLM output: generated captions
2s: i got my barbecue shoes on 9s: go down to the house the sausage 14s: i got my barbecue shoes 16s: welcome to the barbecue pit boys 18s: hi 18s: and today we're gonna do some country meatballs and gravy on the grill 22s: all right 23s: it's real easy to get started here 25s: we got some large granulated salt we got some black peppercorn there we have some granulated onion here 31s: we got some red pepper flakes and here we got some granulated garlic 36s: now instead of using those spices you could just say use your favorite blend of seasonings here ...	2s: i got my barbecue shoes on 9s: go down to the house the sausage 14s: i got my barbecue shoes 16s: welcome to the barbecue pit boys 18s: hi 18s: and today we're gonna do some country meatballs and gravy on the grill 22s: all right 23s: it's real easy to get started here 25s: we got some large granulated salt we got some black peppercorn there we have some granulated onion here 31s: we got some red pepper flakes and here we got some granulated garlic 36s: now instead of using those spices you could just say use your favorite blend of seasonings here ...
426s: you need three out of five 432s: so first jack jack 435s: ready 436s: oh good man 437s: oh no 438s: come on 448s: everything 449s: here we go 449s: okay actually okay 450s: and then for - 451s: since you're a little older you guys stand behind the line here 456s: so let's have you stand behind jack jack's face ...	426s: A person announces that three out of five is needed. 432s: Jack Jack is mentioned. 435s: Someone says "ready." 436s: A person says "oh good man." 437s: A person says "oh no." 438s: Someone says "come on." 448s: Everything is mentioned. 449s: The current video segment starts. 450s: The game is continued. 451s: A person tells the players to stand behind the line. 456s: Jack Jack stands behind his face. 462s: The game continues. 463s: Someone tells Jack Jack to stand up. ...
3s: so we got to the campground 7s: we want to use the oven 10s: we need to light the manual pilot 11s: we only have to light it once for the whole weekend if we're not moving around 15s: so all we need to do is make sure the propane works the oven works or the stovetop works 22s: now we just need to use the oven 24s: so now we're down at the ovens level 26s: we're going to light the pilot light so all we need to do is have a stick lighter 35s: turn the knob to pilot push and hold light 41s: the pilot the pilots burning one two three i let go didn't turn the knob the pilots staying lit 49s: the pilot is good for all weekend 51s: when you go to cook your pizza set it to the temperature you want pizzas done turn it back to pilot and the pilot will stay lit all weekend 63s: when you're ready to leave then we turn it to off that turns off the pilot 69s: we don't want that pilot on while we're driving all the time and constantly using propane 75s: that's why they give us the option 80s: pilot or off is off	3s: Campground 7s: Oven 10s: Manual pilot light 11s: Light pilot once for the weekend 15s: Check if propane and oven work 22s: Use the oven 24s: At oven's level 26s: Light pilot 35s: Turn knob to pilot, push and hold 41s: Pilot lit 49s: Good for all weekend 51s: Set temperature, turn to pilot, and leave 63s: Turn off when ready to leave 69s: Off while driving 75s: Option to choose pilot or off 80s: Off is off. Summary: A group checks if the campground's oven works and lights the manual pilot. They only need to light it once for the weekend as long as they're not moving around. They use the oven and make sure the propane works before they leave. The option to choose pilot or off is given. The pilot should be off while driving.

Table E.10: **Illustrative failures** in caption generation with the Vicuna-13B. Input ASR subtitles and output generated captions where the Vicuna-13B failed to generate video descriptions based on the subtitles. Failures include 1) direct input repetition; 2) ineffective reformulation of the ASR subtitles into descriptions using structures like “A person says ..”; 3) failure to follow the requested structure “a timestamp: a sentence”. For brevity, lengthy examples are shortened with the symbol “...”.

APPENDIX – UNBIASING THROUGH TEXTUAL DESCRIPTIONS: MITIGATING REPRESENTATION BIAS IN VIDEO BENCHMARKS

This appendix contains supplementary material related to Chapter 8, *Unbiasing through Textual Descriptions: Mitigating Representation Bias in Video Benchmarks*, of this thesis. We first present additional experimental results, including extended results on benchmarking video models and common-sense bias, ablation studies, as well as analysis of class distribution in UTD-splits in Section F.1. Next, we provide further implementation details of our UTD method in Section F.2. We then demonstrate qualitative examples from our UTD-descriptions and UTD-splits datasets, along with samples from the user study, in Section F.3. Finally, we discuss the UTD dataset’s license in Section F.4, and the limitations of our work and its broader impact in Section F.5.

Contents

F.1	Additional Results	257
F.1.1	Benchmarking Video Models	257
F.1.2	Common Sense Bias	258
F.1.3	Ablation Study	258
F.1.4	Class Distribution in UTD-Splits	260
F.2	Additional Implementation Details	262
F.2.1	Obtaining Textual Descriptions	262
F.2.2	Getting Text Embeddings	262
F.2.3	Unbiasing Datasets	262
F.2.4	Benchmarking Video Models	264
F.3	Qualitative Results	264
F.4	License Information	265
F.5	Limitations and Broader Impact	265

F.1 ADDITIONAL RESULTS

F.1.1 Benchmarking Video Models

In this section, we present additional benchmarking results for state-of-the-art video models on our object-debiased UTD-splits. Specifically, we extend the analysis presented in Tables 8.5 and 8.6 of the main part of the thesis by including three considered classification datasets, namely Kinetics 600, Kinetics 700, and MiT, and three considered retrieval datasets, namely LSMDC, YouCook2, and Spoken-MiT. In Table F.1, we provide the performance of selected video models on all classification datasets, evaluated both on the full test/val sets and on our debiased UTD-splits. The models were chosen based on their strong performance in Table 8.5 of the main part of the thesis. And in Table F.2, we present the evaluation results for video models across all considered retrieval datasets.

	UCF- split	UCF- UTD- split	UCF- s. bal.	SSv2- split	SSv2- UTD- split	SSv2- s. bal.	K400- split	K400- UTD- split	K400- s. bal.	K600- split	K600- UTD- split	K600- s. bal.	K700- split	K700- UTD- split	K700- s. bal.	MiT- split	MiT- UTD- split	MiT- s. bal.
videomae-L-UH	95.7	90.9 _{-4.8}	91.0 _{-4.7}	67.3	65.4 _{-1.9}	66.2 _{-1.1}	75.8	59.3 _{-16.5}	63.3 _{-12.5}	76.9	61.1 _{-15.8}	65.8 _{-11.1}	65.6	47.6 _{-18.6}	52.7 _{-12.9}	38.1	26.4 _{-11.7}	30.8 _{-7.3}
videomae-v2-B-K710-fnK710	99.0	98.1 _{-0.9}	97.6 _{-1.4}	57.1	54.7 _{-2.4}	55.9 _{-1.2}	85.0	71.9 _{-13.1}	75.5 _{-9.5}	85.6	73.2 _{-12.4}	77.2 _{-8.4}	75.9	60.0 _{-15.9}	65.3 _{-10.6}	40.9	28.9 _{-12.0}	33.9 _{-7.0}
allinone-B-WV2M	84.5	63.1 _{-21.2}	72.9 _{-11.6}	26.2	22.5 _{-3.7}	24.7 _{-1.5}	66.9	42.6 _{-24.3}	50.8 _{-16.1}	68.0	45.0 _{-23.0}	53.3 _{-14.7}	55.2	32.1 _{-21.1}	40.0 _{-15.2}	29.9	16.9 _{-13.0}	22.0 _{-7.9}
umt-B-fnK710	99.0	96.8 _{-2.2}	97.1 _{-1.9}	49.4	46.6 _{-2.8}	48.1 _{-1.3}	85.6	72.4 _{-13.2}	76.2 _{-9.4}	86.2	73.8 _{-12.4}	78.1 _{-8.1}	76.9	61.5 _{-15.4}	66.8 _{-10.1}	39.7	27.1 _{-12.6}	32.1 _{-7.6}
umt-L-fnK710	98.9	96.7 _{-2.2}	97.1 _{-1.8}	57.9	55.5 _{-2.4}	56.7 _{-1.2}	89.0	78.7 _{-10.3}	81.6 _{-7.4}	89.0	78.9 _{-10.1}	82.3 _{-6.7}	82.1	69.3 _{-12.8}	74.0 _{-8.1}	44.5	32.3 _{-12.2}	37.2 _{-7.3}
videomamba-vm-25M	94.3	83.0 _{-11.3}	86.5 _{-7.8}	48.7	45.9 _{-2.8}	47.1 _{-1.6}	78.4	59.1 _{-19.3}	64.6 _{-13.8}	78.5	60.2 _{-18.3}	66.6 _{-11.9}	68.1	47.5 _{-20.8}	54.5 _{-13.6}	37.9	24.5 _{-13.4}	29.7 _{-8.2}
internvid-B-10M-FLT	94.0	81.1 _{-12.9}	84.7 _{-9.3}	48.1	45.2 _{-2.9}	46.7 _{-1.4}	78.6	59.3 _{-19.3}	64.9 _{-13.7}	78.7	60.1 _{-18.6}	66.6 _{-12.1}	68.5	47.8 _{-20.7}	55.0 _{-13.5}	39.3	26.0 _{-13.3}	31.1 _{-8.2}
internvid-B-200M	94.5	82.9 _{-11.6}	85.2 _{-9.3}	54.4	51.8 _{-2.6}	52.8 _{-1.6}	80.0	61.7 _{-18.3}	66.8 _{-13.2}	79.9	62.0 _{-17.9}	68.5 _{-11.4}	70.2	50.3 _{-19.9}	57.6 _{-12.6}	39.9	26.6 _{-13.3}	31.9 _{-8.0}
internvid-L-10M-FLT	95.5	86.3 _{-9.2}	88.4 _{-7.1}	53.6	50.9 _{-2.7}	52.1 _{-1.5}	81.9	64.7 _{-17.2}	69.6 _{-12.3}	81.5	64.8 _{-16.7}	70.8 _{-10.7}	72.4	53.3 _{-19.1}	60.3 _{-12.1}	41.9	28.9 _{-13.0}	33.8 _{-8.1}
internvid-L-200M	96.8	90.5 _{-6.3}	91.9 _{-4.9}	63.1	61.0 _{-2.1}	61.9 _{-1.2}	84.6	69.4 _{-15.2}	73.9 _{-10.7}	84.3	69.3 _{-15.0}	75.1 _{-9.2}	76.4	59.4 _{-17.0}	66.0 _{-10.4}	44.3	31.7 _{-12.6}	36.9 _{-7.4}

Table F.1: **Benchmarking video models in action classification on all six considered classification datasets.** We report accuracy on full test/val sets and our debiased UTD-splits and UTD-splits-balanced (UTD-s.balan.) The accuracy differences with respect to the full test/val sets are color-coded.

	MSR	MSR- UTD- split	DDM	DDM- UTD- split	ANet	ANet- UTD- split	LSMDC	LSMDC- UTD- split	YC2	YC2- UTD- split	S-MiT	S-MiT- UTD- split
umt-b-5M	30.0	20.5 _{-9.5}	30.2	20.6 _{-9.6}	28.6	22.3 _{-6.3}	14.1	9.2 _{-4.9}	6.1	5.2 _{-0.9}	47.9	34.0 _{-13.9}
umt-b-17M	35.6	26.3 _{-9.3}	37.7	27.6 _{-10.1}	34.2	27.3 _{-6.9}	16.6	11.1 _{-5.5}	8.4	6.9 _{-1.5}	53.5	39.5 _{-14.0}
umt-b-25M	35.3	24.8 _{-10.5}	34.2	24.6 _{-9.6}	25.1	19.7 _{-5.4}	13.1	9.0 _{-4.1}	10.3	8.7 _{-1.6}	53.9	40.2 _{-13.7}
umt-l-5M	34.8	24.9 _{-9.9}	33.5	21.8 _{-11.7}	34.8	28.7 _{-6.1}	21.5	16.5 _{-5.0}	7.1	5.8 _{-1.3}	51.9	37.9 _{-14.0}
umt-l-17M	43.6	31.1 _{-12.5}	46.3	35.6 _{-10.7}	45.9	38.7 _{-7.2}	21.6	16.7 _{-4.9}	14.4	11.9 _{-2.5}	60.7	46.9 _{-13.8}
umt-l-25M	42.3	30.6 _{-11.7}	43.6	33.5 _{-10.1}	36.7	30.4 _{-6.3}	19.4	14.2 _{-5.2}	15.1	12.8 _{-2.3}	60.8	47.9 _{-12.9}
videomamba-vm-5M	33.3	23.0 _{-10.3}	37.1	27.1 _{-10.0}	37.1	30.1 _{-7.0}	17.6	12.7 _{-4.9}	6.5	5.6 _{-0.9}	47.6	34.0 _{-13.6}
videomamba-vm-17M	34.9	25.5 _{-9.4}	40.6	28.9 _{-11.7}	40.4	33.0 _{-7.4}	20.1	15.5 _{-4.6}	7.7	6.6 _{-1.1}	51.6	38.3 _{-13.3}
videomamba-vm-25M	34.9	25.5 _{-9.4}	41.4	30.5 _{-10.9}	41.1	33.8 _{-7.3}	20.4	15.4 _{-5.0}	9.3	7.9 _{-1.4}	53.2	39.7 _{-13.5}
internvid-B-10M-FLT	37.9	25.4 _{-12.5}	28.6	17.2 _{-11.4}	24.4	18.8 _{-5.6}	17.0	10.7 _{-6.3}	8.1	5.9 _{-2.2}	48.9	34.6 _{-14.3}
internvid-B-200M	38.1	24.7 _{-13.4}	30.2	19.4 _{-10.8}	26.2	20.1 _{-6.1}	18.3	11.9 _{-6.4}	8.6	6.4 _{-2.2}	49.8	35.8 _{-14.0}
internvid-L-10M	26.7	18.7 _{-8.0}	22.6	15.6 _{-7.0}	21.5	16.3 _{-5.2}	11.4	7.0 _{-4.4}	6.8	5.5 _{-1.3}	35.9	23.6 _{-12.3}
internvid-L-WV10M	26.5	17.6 _{-8.9}	22.2	13.8 _{-8.4}	23.1	17.5 _{-5.6}	12.3	7.6 _{-4.7}	6.7	5.6 _{-1.1}	38.8	25.9 _{-12.9}
internvid-L-10M-DIV	37.3	24.2 _{-13.1}	26.9	15.9 _{-11.0}	23.2	17.7 _{-5.5}	15.7	10.3 _{-5.4}	8.6	6.7 _{-1.9}	48.9	34.4 _{-14.5}
internvid-L-10M-FLT	38.7	26.3 _{-12.4}	29.2	18.8 _{-10.4}	24.5	19.0 _{-5.5}	19.5	13.5 _{-6.0}	9.4	7.5 _{-1.9}	50.5	36.2 _{-14.3}
internvid-L-50M	32.4	22.0 _{-10.4}	26.5	18.3 _{-8.2}	24.4	18.0 _{-6.4}	17.8	11.7 _{-6.1}	8.0	6.4 _{-1.6}	45.7	31.5 _{-14.2}
internvid-L-200M	38.2	24.8 _{-13.4}	30.3	20.2 _{-10.1}	28.7	22.1 _{-6.6}	20.1	13.7 _{-6.4}	11.0	8.8 _{-2.2}	53.7	39.0 _{-14.7}

Table F.2: **Benchmarking video-language models in text-to-video retrieval on all six considered retrieval datasets.** We report accuracy on full test/val sets and our debiased UTD-splits. The accuracy differences with respect to the full test/val sets are color-coded.

F.1.2 Common Sense Bias

In Table F.3, we present additional common sense bias results for all 16 conceptual-temporal combinations across the 12 datasets considered. The observed effects align with results discussed in the main part of the thesis. Specifically, the overall classification performance drops only slightly when predictions are based solely on objects compared to the objects+composition+activities setup across most datasets.

F.1.3 Ablation Study

As discussed in the main part of the thesis, to estimate representation bias, we design a strong model that performs action classification and text-to-video retrieval based solely on the textual descriptions of videos. Throughout our pipeline, we utilize state-of-the-art models, namely

Action Classification Datasets												
	UCF				SSv2				K400			
	seq.-of-f.	avg.-over-f.	max-score-f.	middle f.	seq.-of-f.	avg.-over-f.	max-score-f.	middle f.	seq.-of-f.	avg.-over-f.	max-score-f.	middle f.
obj+comp+act	66.3	66.7 _{+0.4}	66.5 _{+0.2}	61.3 _{-5.0}	6.4	6.8 _{+0.4}	7.4 _{+1.0}	6.0 _{-0.4}	48.0	46.6 _{-1.4}	48.0	39.7 _{-8.3}
objects	63.3 _{-3.0}	61.9 _{-4.4}	62.5 _{-3.8}	57.4 _{-8.9}	5.3 _{-1.1}	4.6 _{-1.8}	5.1 _{-1.3}	4.0 _{-2.4}	45.9 _{-2.1}	41.0 _{-7.0}	42.3 _{-5.7}	35.0 _{-13.0}
activities	67.4 _{+1.1}	67.4 _{+1.1}	65.7 _{-0.6}	59.0 _{-7.3}	6.4	6.0 _{-0.4}	6.0 _{-0.4}	4.9 _{-1.5}	45.2 _{-2.8}	42.1 _{-5.9}	42.8 _{-5.2}	31.3 _{-16.7}
verbs	50.8 _{-15.5}	41.5 _{-24.8}	43.7 _{-22.6}	31.1 _{-35.2}	5.8 _{-0.6}	4.4 _{-2.0}	4.1 _{-2.3}	3.1 _{-3.3}	24.8 _{-23.2}	16.1 _{-31.9}	20.1 _{-27.9}	10.9 _{-37.1}
	K600				K700				MiT			
	seq.-of-f.	avg.-over-f.	max-score-f.	middle f.	seq.-of-f.	avg.-over-f.	max-score-f.	middle f.	seq.-of-f.	avg.-over-f.	max-score-f.	middle f.
obj+comp+act	44.1	42.1 _{-2.0}	43.3 _{-0.8}	35.6 _{-8.5}	39.0	37.1 _{-1.9}	38.7 _{-0.3}	31.1 _{-7.9}	22.6	23.4 _{+0.8}	21.4 _{-1.2}	20.1 _{-2.5}
objects	41.8 _{-2.3}	36.5 _{-7.6}	37.5 _{-6.6}	31.0 _{-13.1}	37.0 _{-2.0}	32.2 _{-6.8}	33.3 _{-5.7}	26.7 _{-12.3}	21.0 _{-1.6}	19.4 _{-3.2}	20.1 _{-2.5}	17.6 _{-5.0}
activities	41.4 _{-2.7}	38.1 _{-6.0}	38.5 _{-5.6}	28.1 _{-16.0}	36.7 _{-2.3}	33.0 _{-6.0}	33.8 _{-5.2}	24.0 _{-15.0}	21.0 _{-1.6}	20.0 _{-2.6}	20.1 _{-2.5}	15.6 _{-7.0}
verbs	21.4 _{-22.7}	13.8 _{-30.3}	16.7 _{-27.4}	8.9 _{-35.2}	17.6 _{-21.4}	11.0 _{-28.0}	13.6 _{-25.4}	7.0 _{-32.0}	16.2 _{-6.4}	12.2 _{-10.4}	14.1 _{-8.5}	9.2 _{-13.4}
Text-to-Video Retrieval Datasets												
	MSR				DDM				ActN			
	seq.-of-f.	avg.-over-f.	max-score-f.	middle f.	seq.-of-f.	avg.-over-f.	max-score-f.	middle f.	seq.-of-f.	avg.-over-f.	max-score-f.	middle f.
obj+comp+act	36.6	31.9 _{-4.7}	33.4 _{-3.2}	23.4 _{-13.2}	27.2	26.5 _{-0.7}	29.1 _{+1.9}	19.8 _{-7.4}	26.6	26.6	21.5 _{-5.1}	13.5 _{-13.1}
objects	32.1 _{-4.5}	28.7 _{-7.9}	29.7 _{-6.9}	17.8 _{-18.8}	27.0 _{-0.2}	26.5 _{-0.7}	25.5 _{-1.7}	16.8 _{-10.4}	24.8 _{-1.8}	22.7 _{-3.9}	17.8 _{-8.8}	11.5 _{-15.1}
activities	25.1 _{-11.5}	22.4 _{-14.2}	18.6 _{-18.0}	11.6 _{-25.0}	21.1 _{-6.1}	21.0 _{-6.2}	18.4 _{-8.8}	11.3 _{-15.9}	21.4 _{-5.2}	17.8 _{-8.8}	13.2 _{-13.4}	8.4 _{-18.2}
verbs	10.5 _{-26.1}	8.7 _{-27.9}	7.8 _{-28.8}	4.2 _{-32.4}	7.0 _{-20.2}	6.0 _{-21.2}	5.5 _{-21.7}	3.5 _{-23.7}	7.4 _{-19.2}	5.1 _{-21.5}	3.9 _{-22.7}	2.5 _{-24.1}
	LSMDC				YC2				S-MiT			
	seq.-of-f.	avg.-over-f.	max-score-f.	middle f.	seq.-of-f.	avg.-over-f.	max-score-f.	middle f.	seq.-of-f.	avg.-over-f.	max-score-f.	middle f.
obj+comp+act	17.0	16.4 _{-0.6}	18.3 _{+1.3}	12.7 _{-4.3}	8.4	8.6 _{+0.2}	8.9 _{+0.5}	6.1 _{-2.3}	45.9	43.8 _{-2.1}	46.1 _{+0.2}	35.5 _{-10.4}
objects	13.6 _{-3.4}	11.7 _{-5.3}	12.3 _{-4.7}	9.2 _{-7.8}	7.9 _{-0.5}	6.2 _{-2.2}	7.1 _{-1.3}	4.9 _{-3.5}	29.8 _{-16.1}	27.5 _{-18.4}	26.4 _{-19.5}	18.2 _{-27.7}
activities	14.7 _{-2.3}	13.6 _{-3.4}	11.4 _{-5.6}	7.8 _{-9.2}	8.1 _{-0.3}	7.3 _{-1.1}	6.3 _{-2.1}	4.1 _{-4.3}	41.1 _{-4.8}	38.4 _{-7.5}	37.8 _{-8.1}	29.3 _{-16.6}
verbs	5.5 _{-11.5}	6.0 _{-11.0}	4.5 _{-12.5}	3.0 _{-14.0}	1.2 _{-7.2}	1.1 _{-7.3}	1.0 _{-7.4}	0.6 _{-7.8}	13.1 _{-32.8}	9.5 _{-36.4}	8.8 _{-37.1}	4.8 _{-41.1}

Table F.3: **Evaluation of common sense bias with respect to all 16 conceptual-temporal combinations on all 12 considered datasets.** We color code with respect to the difference to objects+composition+activities (obj+comp+act) concepts in sequence-of-frames (seq.-of-f.) temporal setup.

Text Embedding Model	Action Classification		Text-to-Video Retrieval	
	UCF	SSv2	MSR	DDM
CLIP [RKH ⁺ 21b] text encoder	51.1	1.8	6.1	7.0
Long-CLIP [ZZD ⁺ 25] text encoder	48.5	2.8	24.0	19.0
E5-Mistral-7B-Instruct [WYH ⁺ 23]	65.6	6.2	35.1	26.7
SFR-Embedding-Mistral [ML] ⁺ 24]	66.3	6.4	36.6	27.2

Table F.4: **Ablation on text embedding model.** Evaluation is performed in objects+composition+activities concepts in sequence-of-frames temporal setup on the full test/-validation splits of the respective datasets. We report accuracy for classification and recall@1 for retrieval. Selected option is bolded.

Vision-Language Model	Action Classification		Text-to-Video Retrieval	
	UCF	SSv2	MSR	DDM
LLaVA-v1.5-7B [LLLL24]	61.8	5.9	35.5	28.7
LLaVA-1.6-Mistral-7B [LLL⁺24]	66.3	6.4	36.6	27.2

Table F.5: **Ablation on vision-language model.** Evaluation is performed in objects+composition+activities concepts in sequence-of-frames temporal setup on the full test/validation splits of the respective datasets. We report accuracy for classification and recall@1 for retrieval. Selected option is bolded.

	Action Classification						Text-to-Video Retrieval					
	UCF	SSv2	K400	K600	K700	MiT	MSR	DDM	ANet	LSMDC	YC2	S-MiT
CLIP ViT-B/32	67.7	1.8	46.7	41.3	34.8	18.6	31.4	26.3	20.4	14.3	4.9	34.7
CLIP ViT-L/14	75.9	2.8	58.7	54.7	48.6	23.7	36.3	29.6	26.2	19.9	8.1	40.4
Ours (obj+comp+act, sequence-of-frames)	66.3	6.4	48.0	44.1	39.0	22.6	36.6	27.2	26.6	17.0	8.4	45.9

Table F.6: **Comparison in zero-shot action classification and text-to-video retrieval.** Note, that our model uses only textual descriptions of video. Evaluation is performed on full test/val splits of the respective datasets. We report accuracy for classification and Recall@1 for retrieval.

LLaVA-1.6-Mistral-7B [LLL⁺24] as the VLM, Mistral-7B-Instruct-v0.2 [JSM⁺23] as the LLM, and SFR-Embedding-Mistral [MLJ⁺24] for text embedding model. In this section, we provide additional analysis of our model.

Text Embedding Model. First, in Table F.4, we compare four strong models for text encoding (χ in Figure 8.3 of the main part of the thesis). Namely, we consider large versions of the CLIP [RKH⁺21b] text encoder and LongCLIP [ZZD⁺25] text encoder, which extends CLIP to better handle long-text inputs. Additionally, we examine two LLM-based text embedding models: E5-Mistral-7B-Instruct [WYH⁺23] and its finetuned version, SFR-Embedding-Mistral [MLJ⁺24], trained on more data. As shown in Table F.4, the text embedding models E5-Mistral-7B-Instruct and SFR-Embedding-Mistral, both pretrained on large text datasets to effectively encode text for tasks such as information retrieval, outperform the CLIP-based text embedding models. In our pipeline, we employ the best-performing SFR-Embedding-Mistral model.

Vision-Language Model. Next, in F.5, we ablate two VLM models for extracting detailed textual descriptions. Specifically, we evaluate LLaVA-v1.5-7B [LLLL24] and LLaVA-1.6-Mistral-7B [LLL⁺24], finding that the latter model achieves the best performance.

Comparison to CLIP. Finally, we evaluate how well our model, which relies solely on textual descriptions, performs for zero-shot video classification and retrieval compared to the CLIP baseline [RKH⁺21b]. The results in Table F.6 demonstrate that our model generally outperforms CLIP ViT-B/32 and performs almost on par with the CLIP ViT-L/14 backbone. This highlights the feasibility of performing action classification and video retrieval based purely on textual descriptions. We further observe that caption-based retrieval performs better compared to caption-based action classification. We attribute this to the fact that captions capture more specific aspects of individual videos, which are also reflected in the generated captions.

F.1.4 Class Distribution in UTD-Splits

In Figure F.1, we show the class distribution in our UTD- and UTD-balanced test/validation splits, in comparison with the original class distribution in the full test/validation splits. We observe that the results align well with our expectations, for example, the class “Apply Eye Makeup” in UCF-101 is significantly reduced in the UTD-split due to a strong object bias.

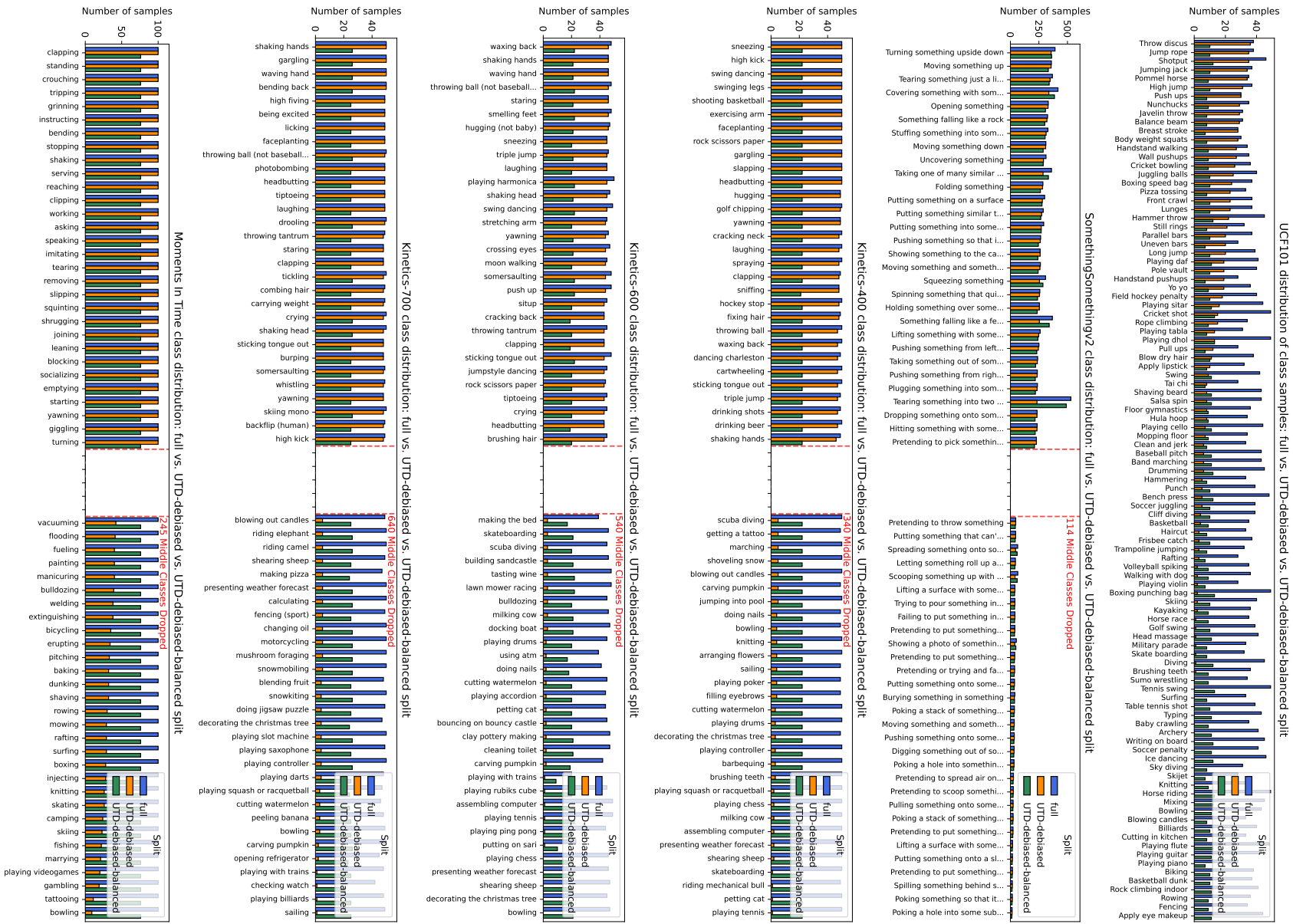


Figure F.1: Comparison of class distribution in full test/val split vs. UTD-debiased split vs. UTD-debiased balanced split for six considered classification datasets.

F.2 ADDITIONAL IMPLEMENTATION DETAILS

In this section, we provide further details about our UTD method.

F.2.1 Obtaining Textual Descriptions

We used a few-shot in-context learning strategy [BMR⁺20] when prompting LLM to extract objects, activities, and verbs from objects+composition+activities descriptions. Namely, we used 3-shots for objects and activities and 5-shots for verbs. We also did simple postprocessing of LLM output. Since the LLM is prompted to output an enumerated list (see prompts in Table F.8), we delete numeration and delete any output in brackets.

We list the prompts used in our work to obtain textual descriptions of frames for different concept categories, namely objects+composition+activities ($d_{n,i}$), objects ($o_{n,i}$), activities ($a_{n,i}$), and verbs ($v_{n,i}$), in Table F.8

F.2.2 Getting Text Embeddings

Following [MLJ⁺24], we used the prompt template “Instruct: <instruction>\nQuery: <input text>” to obtain text embeddings for various input text descriptions using the SFR-Embedding-Mistral model [MLJ⁺24]. In Table F.9, we provide the instructions used for action classification in different settings, and in Table F.10, we present instructions used for text-to-video retrieval.

F.2.3 Unbiasing Datasets

As stated in the main part of the thesis, we debias the test/val sets of the considered datasets by excluding samples that are classified or retrieved correctly ($M(v_n, \phi) = 1$) based on *object sequence-of-frame* representation.

For text-to-video retrieval, we debias datasets with respect to *common-sense bias*. To minimize the impact of random fluctuations in the process, we prompt the text embedding models with three different prompts for query captions and three different prompts for object textual descriptions for videos, generating three embeddings for each query caption and for each video. Prompts are reported in Table F.11. Consequently, we perform zero-shot text-to-video retrieval inference using all nine combinations of query embeddings and video embeddings, resulting in nine predictions. We exclude a text query from the test/val set only if all nine predictions agree on the correct Top-1 retrieval for the corresponding video.

For action classification, we debias datasets with respect to *dataset bias*. Specifically, we generate three different embeddings for the object textual descriptions of each video using three different instructions for the text embedding model (Table F.11), resulting in three sets of video embeddings. Using each set of embeddings, we further train three linear classifiers on bootstrapped training sets (we sample training sets using sampling replacement to match the original training set size). A sample is excluded from the test/val set only if all nine models (across the three embedding sets and three linear models) agree on the correct Top-1 classification. The percentage of removed samples is determined automatically based on the extent of object representation bias in the dataset.

Since debiasing may disproportionately remove certain label classes in classification datasets, we additionally construct balanced UTD splits. While maintaining the total number of removed

Model	Backbone	Pretraining Datasets	Finetuning Datasets
VideoMAE-B-K400 [TSWW22]	ViT-B/16	Kinetics-400 (w/o labels) [KCS ⁺ 17]	-
VideoMAE-B-UH [TSWW22]	ViT-B/16	UnlabeledHybrid [WHZ ⁺ 23]: K700 [CNHZ19] + WebVid2M [BNVZ21a] + SS [GEKM ⁺ 17b] + AVA [GSR ⁺ 18] + Instagram (collected)	-
VideoMAE-L-UH [TSWW22]	ViT-L/14	UnlabeledHybrid [WHZ ⁺ 23]: K700 [CNHZ19] + WebVid2M [BNVZ21a] + SS [GEKM ⁺ 17b] + AVA [GSR ⁺ 18] + Instagram (collected)	-
VideoMAE-H-UH [TSWW22]	ViT-H/16	UnlabeledHybrid [WHZ ⁺ 23]: K700 [CNHZ19] + WebVid2M [BNVZ21a] + SS [GEKM ⁺ 17b] + AVA [GSR ⁺ 18] + Instagram (collected)	-
VideoMAEv2-B-K710-fnK710 [WHZ ⁺ 23]	ViT-B/16	Kinetics-710 [LWH ⁺ 22] (Kinetics-400 [KCS ⁺ 17] + Kinetics-600 [CNB ⁺ 18] + Kinetics-710 [LWH ⁺ 22]) (w/o labels)	-
AllInOne-B-WV2M+CC [WGY ⁺ 23]	ViT-B/16	WebVid2M [BNVZ21a] + CC3M [SDGS18]	-
AllInOne-B-WV2M+HT [WGY ⁺ 23]	ViT-B/16	WebVid2M [BNVZ21a] + HowTo100M [MZA ⁺ 19]	-
AllInOne-B-WV2M+HT+CC+YTT+ [WGY ⁺ 23]	ViT-B/16	WebVid2M [BNVZ21a] + HowTo100M [MZA ⁺ 19] + YTT [ZLH ⁺ 21a] + CC3M [SDGS18] + CC12M [CSDS21] + COCO [LMB ⁺ 14] + VG [KZG ⁺ 17] + SBU [OKB11]	-
UMT-B-K710 [LWL ⁺ 23]	UMT-B/16	Kinetics-710 [LWH ⁺ 22] (w/o labels)	-
UMT-B-fnK710 [LWL ⁺ 23]	UMT-B/16	Kinetics-710 [LWH ⁺ 22] (w/o labels)	Kinetics-710 [LWH ⁺ 22]
UMT-L-K710 [LWL ⁺ 23]	UMT-L/16	Kinetics-710 [LWH ⁺ 22] (w/o labels)	-
UMT-L-fnK710 [LWL ⁺ 23]	UMT-L/16	Kinetics-710 [LWH ⁺ 22] (w/o labels)	Kinetics-710 [LWH ⁺ 22]
UMT-B-5M [LWL ⁺ 23]	UMT-B/16	Kinetics-710 [LWH ⁺ 22] (w/o labels) + WebVid2M [BNVZ21a] + CC3M [SDGS18]	-
UMT-B-17M [LWL ⁺ 23]	UMT-B/16	Kinetics-710 [LWH ⁺ 22] (w/o labels) + WebVid2M [BNVZ21a] + CC3M [SDGS18] + CC12M [CSDS21] + COCO [LMB ⁺ 14] + VG [KZG ⁺ 17] + SBU [OKB11]	-
UMT-B-25M [LWL ⁺ 23]	UMT-B/16	Kinetics-710 [LWH ⁺ 22] (w/o labels) + WebVid10M [BNVZ21a] + CC3M [SDGS18] + CC12M [CSDS21] + COCO [LMB ⁺ 14] + VG [KZG ⁺ 17] + SBU [OKB11]	-
UMT-L-5M [LWL ⁺ 23]	UMT-L/16	Kinetics-710 [LWH ⁺ 22] (w/o labels) + WebVid2M [BNVZ21a] + CC3M [SDGS18]	-
UMT-L-17M [LWL ⁺ 23]	UMT-L/16	Kinetics-710 [LWH ⁺ 22] (w/o labels) + WebVid2M [BNVZ21a] + CC3M [SDGS18] + CC12M [CSDS21] + COCO [LMB ⁺ 14] + VG [KZG ⁺ 17] + SBU [OKB11]	-
UMT-L-25M [LWL ⁺ 23]	UMT-L/16	Kinetics-710 [LWH ⁺ 22] (w/o labels) + WebVid10M [BNVZ21a] + CC3M [SDGS18] + CC12M [CSDS21] + COCO [LMB ⁺ 14] + VG [KZG ⁺ 17] + SBU [OKB11]	-
VideoMamba-VM-K400 [LLW ⁺ 24]	VideoMamba-M	Kinetics-400 [KCS ⁺ 17] (w/o labels)	-
VideoMamba-VM-5M [LLW ⁺ 24]	VideoMamba-M	Kinetics-400 [KCS ⁺ 17] (w/o labels) + WebVid2M [BNVZ21a] + CC3M [SDGS18]	-
VideoMamba-VM-17M [LLW ⁺ 24]	VideoMamba-M	Kinetics-400 [KCS ⁺ 17] (w/o labels) + WebVid2M [BNVZ21a] + CC3M [SDGS18] + CC12M [CSDS21] + COCO [LMB ⁺ 14] + VG [KZG ⁺ 17] + SBU [OKB11]	-
VideoMamba-VM-25M [LLW ⁺ 24]	VideoMamba-M	Kinetics-400 [KCS ⁺ 17] (w/o labels) + WebVid10M [BNVZ21a] + CC3M [SDGS18] + CC12M [CSDS21] + COCO [LMB ⁺ 14] + VG [KZG ⁺ 17] + SBU [OKB11]	-
InternVid-B-10M-FLT [WHL ⁺ 23]	ViCLIP-B/16	InternVid-10M-FLT [WHL ⁺ 23]	-
InternVid-B-200M [WHL ⁺ 23]	ViCLIP-B/16	InternVid-200M [WHL ⁺ 23]	-
InternVid-L-10M [WHL ⁺ 23]	ViCLIP-L/14	InternVid-10M [WHL ⁺ 23]	-
InternVid-L-WebVid10M [WHL ⁺ 23]	ViCLIP-L/14	WebVid10M [BNVZ21a]	-
InternVid-L-10M-DIV [WHL ⁺ 23]	ViCLIP-L/14	InternVid-10M-DIV [WHL ⁺ 23]	-
InternVid-L-10M-FLT [WHL ⁺ 23]	ViCLIP-L/14	InternVid-10M-FLT [WHL ⁺ 23]	-
InternVid-L-50M [WHL ⁺ 23]	ViCLIP-L/14	InternVid-50M [WHL ⁺ 23]	-
InternVid-L-200M [WHL ⁺ 23]	ViCLIP-L/14	InternVid-200M [WHL ⁺ 23]	-

Table F.7: Overview of the 30 models analyzed in this paper, including their backbones, pretraining datasets, and supervised finetuning datasets.

samples as in the original debiasing method, we adjust the number of samples removed from each class based on their average confidence (across nine models) to preserve the original class proportions.

Reliability of filtering biased samples. To summarize, we ensure robust and reliable filtering of biased samples by leveraging state-of-the-art VLMs and LLMs, applying prompt engineering, and employing an in-context learning strategy to extract specific concepts (e.g., objects) while minimizing leakage from unrelated concepts. We also conduct a user study to validate the reliability of the extracted concepts. To further mitigate false positives, we aggregate predictions across nine different prompt/model combinations.

F.2.4 Benchmarking Video Models

Table F.7 summarizes all 30 video models evaluated in this work, detailing their architectural backbones, pretraining datasets, and supervised finetuning setups.

Further, we provide additional implementation details on our evaluation setup of video backbones in action classification and text-to-video retrieval. To evaluate a video backbone in action classification, following the VideoGLUE benchmark [YGZ⁺23], we train a classification model with the corresponding frozen video backbone and single-layer pooler head [YGZ⁺23] with one classification linear layer as described in the main part of the thesis. For training and evaluation, we use 8 uniformly sample frames as inputs, however, for video backbones that create 3D tokens with a stride over frames, such as VideoMAE [TSWW22, WHZ⁺23], we respectively scale the number of input frames, namely, we use 16 frames for VideoMAE and VideoMAEv2, to ensure that models use the same number of tokens for the same model size. We train a model for 50 epochs with an AdamW optimizer [LH19] and a learning rate of 0.001. We use a cosine weight decay scheduler with five epochs warmup. We follow the augmentation pipeline of the VideoGLUE benchmark [YGZ⁺23]. We do not use multi-crop evaluation to simplify and standardize the evaluation setting for all datasets.

For text-to-video retrieval, we evaluate the zero-shot capabilities of the text-video models and use the respective backbones without finetuning. Same as in action classification, we use 8 uniformly sampled frames as inputs. We also follow the corresponding model recipes in using re-ranking, namely, we rerank 128 videos with highest similarities based on dual encoder output by a joint encoder for Unmasked Teacher (UMT) [LWL⁺23] and VideoMamba [LLW⁺24] models.

F.3 QUALITATIVE RESULTS

UTD-descriptions. In Figures F.2 to F.5, we present qualitative results of textual descriptions of our *UTD-descriptions* dataset for different concept categories using random videos from the MSRVT dataset. We observed that the VLM provides detailed descriptions of objects+composition+activities (Figure F.2). Furthermore, the LLM successfully parses these descriptions into objects, activities, and verbs (Figures F.3 to F.5).

UTD-splits. In Figure F.6, we present qualitative examples from the full test set and our object-debiased UTD-split on the UCF101 dataset. We observe that samples from our object-debiased UTD-split demand a deeper level of video understanding beyond simple object recognition. For instance, in examples of classes involving playing musical instruments, such as “Playing Daf”, “Playing Cello”, or “Playing Sitar”, the videos often include additional instruments in the background, such as a piano and drums in the case of “Playing Daf” example, alongside the primary instrument. Similarly, in the “Pizza Tossing” example, the pizza in the UTD-split example is barely visible, and a video requires analysis beyond this single frame for correct class prediction.

User Study. As stated in the main part of the thesis, our user study shows that 87.6% (667 out of 761) of the VLM-recognized objects are identified as visible, with only 94 objects not selected as visible. To better understand the VLM’s errors, we manually classify these 94 objects into five categories: 1) attribute error (e.g., the object is “right hand” instead of “left hand”), 2) misclassification (the object is present but incorrectly identified as a different object, e.g., a “snowboard” instead of “snow slide”), 3) hallucination, 4) human annotation mistake – the object is visible, and 5) other. In Figure F.7 we provide examples from these five categories.

F.4 LICENSE INFORMATION

Our *UTD-descriptions* are generated by LLaVA-1.6-7B-mistral model [LLWL23b, LLL⁺24] as well as Mistral-7B-Instruct-v0.2 [JSM⁺23]. The models were used according to their licenses. LLaVA-1.6-7B-mistral model complies with the base LLM’s model license¹, which is Mistral-7B. Mistral-7B model follows Apache 2.0 license². We release our *UTD dataset*, comprising *UTD-descriptions* and *UTD-splits*, under the CC BY 4.0 license. However, specific components of the underlying dataset may be governed by stricter licensing conditions from the corresponding video datasets.

F.5 LIMITATIONS AND BROADER IMPACT

Limitations. Due to the high human annotation cost of detailed frame text descriptions for videos, our method of evaluating and debiasing datasets relies on the textual description of video frames generated by VLMs. Therefore, these textual descriptions may potentially misrepresent the video due to the potential limitation of VLM models. Namely, such textual description might be prone to hallucinations, namely describing things that are not present in the frame, having social biases, such as guessing a person’s occupation based on how the person looks or implying some information that is not visible. Then, we further extract different concept categories from these textual descriptions using LLM. These steps might be prone to leaking information about other concepts, such as adding activity information to the objects list. Therefore, the quality of our textual descriptions with respect to different concepts also limited the filtering abilities of LLMs. In our user study evaluating the quality of generated descriptions of objects, we found the hallucination rate to be approximately 6%. Our biased discovering method is also limited by the performance of pretrained text embedding models, and using stronger text embedding models might lead to discovering even stronger biases in the datasets, such as more videos being classified correctly based on only objects using common sense.

Potential Negative Societal Impact. In our work, we perform bias analysis of the existing video classification and text-to-video retrieval datasets, 12 datasets in total, as well as provide debiased evaluation splits – *UTD-splits*. Since our annotations are based on existing datasets, our data distribution reflects the social biases inherent in those sources. We also provide *UTD-descriptions* for these 12 datasets – textual descriptions of objects, activities, verbs, and objects+composition+activities categories. As discussed in the limitations, these descriptions might be prone to social biases potentially present in the current VLM models. Therefore, such descriptions might potentially propagate these social biases. To create the dataset, we utilize VLM and LLM models, which contribute to increased energy consumption and carbon emissions as a negative externality.

¹https://github.com/haotian-liu/LLaVA/blob/main/docs/MODEL_ZOO.md

²<https://mistral.ai/news/announcing-mistral-7b/>

	Prompt
Prompting LLaVA-1.6-Mistral-7B: Obtaining Objects+Composition+Activities $d_{n,i}$	Describe the objects relationships in the photo.
Prompting Mistral-7B-Instruct-v0.2: Obtaining Objects $o_{n,i}$	<s>[INST] You are an intelligent chatbot designed to extract requested information from the textual description of an image. I will give you a textual description of the image. List ALL objects visible in the image. An object is anything that has a fixed shape or form, that you can touch or see. Name each object with one noun or a maximum of two words. Skip uncertain objects. The textual description of the image: "<INPUT TEXTUAL DESCRIPTION>" DO NOT PROVIDE ANY EXTRA INFORMATION ABOUT OBJECT PROPERTIES OR RELATIONSHIPS TO OTHER OBJECTS IN PARENTHESES. DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION. [/INST] Comprehensive enumerated list of objects:
Prompting Mistral-7B-Instruct-v0.2: Obtaining Activities $a_{n,i}$	<s>[INST] You are an intelligent chatbot designed to extract requested information from the textual description of an image. I will give you a textual description of the image. List all VISIBLE activities in the image. Activity is lively action or movement. Name each activity with a concise phrase SKIP possible or implied activities that are not visible . If no activity is visible, reply "No activity is visible." DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION. The textual description of the image: "<INPUT TEXTUAL DESCRIPTION>" [/INST] Comprehensive enumerated list of activities:
Prompting Mistral-7B-Instruct-v0.2: Obtaining Verbs $v_{n,i}$	<s>[INST] You are an intelligent chatbot designed to extract requested information from the textual description of an image. I will give you a list of visible activities of the image. Your task is to delete information about objects from this description. Replace all objects in this list with "someone" or "something," but keep the activity. If you have to, you may delete some details, but delete ALL object information. If the input is "No activity is visible .", keep it "No activity is visible." DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION. The list of visible activities: "<INPUT ACTIVITIES DESCRIPTION>" [/INST] Post-processed enumerated list of activities:
Prompting Mistral-7B-Instruct-v0.2: Obtaining 15-words Summaries $d'_{n,i}$	<s>[INST] You are an intelligent chatbot designed to extract requested information from the textual description of an image. Summarize the following image description in 15 words: "<INPUT TEXTUAL DESCRIPTION>" [/INST] 15-words summary:

Table F.8: Prompts used in our UTD method to obtain textual descriptions of frames with respect to different concepts categories: objects+composition+activities $d_{n,i}$, objects $o_{n,i}$, activities $a_{n,i}$, and verbs $v_{n,i}$.

Textual description	Setup	<instruction>
objects+composition+activities $d_{n,i}$	single-frame	Given a video frame description, retrieve the activity depicted in this video.
	sequence-of-frames	Given descriptions of video frames, retrieve the activity depicted in this video.
objects $o_{n,i}$	single-frame	Given a list of objects visible on the video frame, retrieve the activity depicted in this video.
	sequence-of-frames	Given lists of objects visible on the video frames, retrieve the activity depicted in this video.
activities $a_{n,i}$ / verbs $v_{n,i}$	single-frame	Given a description of actions visible on the video frame, retrieve the activity depicted in this video.
	sequence-of-frames	Given a description of actions visible on the video frames, retrieve the activity depicted in this video.
activity class name	single-frame	Given an activity, retrieve a video frame description that may depict this activity.
	sequence-of-frames	Given an activity, retrieve a video description that may depict this activity.

Table F.9: Instructions used to prompt the SFR-Embedding-Mistral model for action classification.

Textual description	Setup	<instruction>
objects+composition+activities $d_{n,i}$	single-frame	Given a description of a single video frame, retrieve a short description of the full video.
	sequence-of-frames	Given descriptions of video frames, retrieve a short description of the full video.
objects $o_{n,i}$	single-frame	Given a list of objects visible on the video frame, retrieve a short video description.
	sequence-of-frames	Given lists of objects visible on the video frames, retrieve a short video description.
activities $a_{n,i}$ / verbs $v_{n,i}$	single-frame	Given a description of actions visible on the video frame, retrieve a short video description.
	sequence-of-frames	Given a description of actions visible on the video frames, retrieve a short video description.
caption (from dataset)	single-frame	Given a short video description, retrieve a description of a specific frame within that video.
	sequence-of-frames	Given a short video description, retrieve another description of this video.

Table F.10: Instructions used to prompt the SFR-Embedding-Mistral model for text-to-video retrieval.

	Objects+composition+activities: In the image, there is a person standing on the bow of a small boat that is floating on a body of water. The person appears to be engaged in an activity that involves a blue, translucent object that is being pulled or dragged by the boat. The object is connected to the boat by a line, and the person seems to be controlling the movement of the boat, which in turn affects the movement of the object. The background shows a calm water surface with trees and a clear sky, suggesting a peaceful, outdoor setting. The text overlay on the image provides information about the person and the activity, indicating that the person is involved in "Authentic Low Country Cooking" and that the image is from a website called "gullahgrubs.com." The overall scene suggests a leisurely activity related to cooking, possibly involving the use of the blue object for food preparation or demonstration.
	Objects+composition+activities: In the image, there is an animated dog character standing next to a yellow bucket. The dog appears to be looking at the bucket, which has a label on it. The dog is wearing a red collar with a tag, and it has a black and white coat with a white muzzle. The background suggests an indoor setting with a wooden floor and a blurred object that could be a bottle or a container in the background. The dog's position relative to the bucket is such that it seems to be interacting with the contents of the bucket, possibly trying to get something out of it. The label on the bucket is not clearly legible, so its contents are not identifiable from the image.
	Objects+composition+activities: In the photo, there is a person's hand holding a black object with a green label. The object appears to be a bottle or container of some sort. The hand is positioned in such a way that it is holding the object securely. The background is blurred, but it seems to be a kitchen setting, as suggested by the presence of what looks like a kitchen counter or table. The focus of the image is on the hand and the object it is holding, with the background being out of focus. The relationship between the objects is that the hand is interacting with the object, possibly opening it or preparing to use it.
	Objects+composition+activities: The image contains two lines of text. The first line reads "An easy folding stroller" and the second line reads "will make." The text is presented in a simple, sans-serif font, and the color of the text is a shade of blue. The relationship between the two lines of text is that the first line is a statement, and the second line is a continuation of the thought or instruction implied by the first line. The second line is a verb, suggesting an action that will be taken or the result of using the stroller mentioned in the first line. The text is centered on the page, and there are no other objects or relationships to describe.
	Objects+composition+activities: The image shows a bowl of food that appears to be a type of noodle dish, possibly a Singaporean hawker meal. The dish is garnished with various ingredients, including slices of what could be a type of seafood, such as shrimp or prawns, green onions, and what looks like fried shallots or garlic. The noodles are likely to be a type of egg noodle, given their appearance and the context of the dish. The ingredients are arranged in a way that suggests a balance of flavors and textures. The seafood provides a protein source and a hint of the sea, while the green onions add a fresh, herbal flavor. The fried shallots or garlic contribute a savory, slightly sweet element to the dish. The dish is presented in a white bowl, which contrasts with the colorful ingredients, making the food look more appetizing. The bowl is placed on a surface that is not clearly visible, but it appears to be a table or countertop. The text overlay on the image provides a description of the dish, indicating that it is a popular Singaporean hawker meal. This suggests that the image is likely intended for a culinary context, such as a food blog or a recipe sharing platform. The text is in English and is superimposed over the image, ensuring that the viewer can read it regardless of the image's resolution or size.
	Objects+composition+activities: In the photo, there is a person sitting at a table with various crafting supplies. The person appears to be engaged in a crafting activity, possibly sewing or working with fabric. The table is cluttered with several items, including: 1. A sewing machine, which is likely being used by the person. 2. A spool of thread, which is being used by the person. 3. A pair of scissors, which are commonly used in sewing and crafting. 4. A roll of tape, which could be used for measuring or securing items. 5. A pair of pliers, which might be used for cutting or bending materials. 6. A bottle of water, which is likely there for the person to drink. 7. A roll of paper towels, which could be used for cleaning up any messes. 8. A pair of tweezers, which might be used for picking up small items or threads. 9. A pair of scissors with orange handles, which could be a different type of scissors or a different color for a specific purpose. 10. A small blue object, which could be a tool or a small part of the crafting project. The objects are arranged in a way that suggests an active workspace, with the person using the sewing machine and the other items within easy reach. The items are not neatly organized, indicating that the person is in the middle of a project and has not yet tidied up the workspace.

Figure F.2: Qualitative examples of objects+composition+activities textual descriptions for random videos in MSRVTT dataset.

**Objects:**

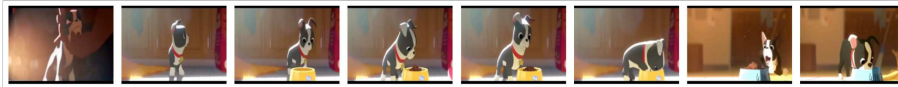
Frame 1: person, small boat, body of water, blue object, string or line
 Frame 2: person, small boat, bow, blue, translucent object, string, wide-brimmed hat, patterned shirt, body of water, trees, blue smoke or vapor, text overlay "gullah grub", "authentic low country cooking", "www
 Frame 3: person, hat, small boat, bow, body of water, blue object, trees, clear sky.
 Frame 4: person, small boat, bow, body of water, blue, translucent object, trees, clear sky.
 Frame 5: person, small boat, body of water, blue, translucent object, line, trees, clear sky, text overlay, website name "gullahgrubs.com"
 Frame 6: person, hat, small boat, body of water, bird, string or line
 Frame 7: person, small boat, bow, fishing rod, line, body of water, trees, clear sky, text: "produced & directed by:", text: "www.outsidetheboxmedia.com", text: "copyright 2011"
 Frame 8: person, hat, small boat, body of water, bird, string or line.

Activities:

Frame 1: A person is holding a blue object. The person is pulling on a string or line attached to the blue object.
 Frame 2: A person is controlling the movement of a blue, translucent object using a string.
 Frame 3: A person is standing on a boat. A person is pulling a blue object through water.
 Frame 4: A person is standing on the bow of a boat. A person is spinning or twirling a blue, translucent object.
 Frame 5: A person is controlling a boat. An object is being pulled or dragged by the boat.
 Frame 6: A person is controlling a bird's flight with a string or line.
 Frame 7: A person is standing on the bow of a boat. A person is fishing or participating in a water-based activity using a fishing rod.
 Frame 8: A person is controlling a bird's flight with a string or line.

Verbs:

Frame 1: Someone is holding something. Someone is pulling on something.
 Frame 2: Someone is moving something using something.
 Frame 3: Someone is standing on something. Someone is pulling something through water.
 Frame 4: Someone is standing. Someone is spinning or twirling something.
 Frame 5: Someone is controlling something. Something is being pulled or dragged.
 Frame 6: Someone is controlling something.
 Frame 7: Someone is standing. Someone is participating in an activity using a rod.
 Frame 8: Someone is controlling something.

**Objects:**

Frame 1: character, red and white striped shirt, room, warm, golden light, fireplace or glowing light source
 Frame 2: animated character, dog, wooden door, red object, bottle or container, label, bell
 Frame 3: animated dog, yellow bucket, wooden floor, red collar, red object
 Frame 4: 3d animated dog, yellow bucket, surface, blurred background, wooden door, red object
 Frame 5: animated dog, yellow bucket, red collar, tag, wooden floor, blurred object
 Frame 6: cartoon dog, yellow object, container or box, indoor setting, wooden surface, red object, bottle or container.
 Frame 7: cartoon dog, blue bowl, wooden door, window, curtains.
 Frame 8: dog, blue bowl, wooden floor, furniture

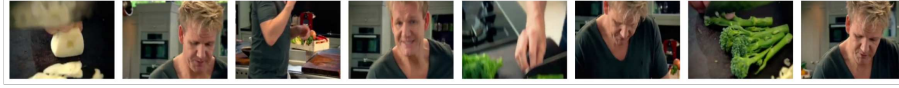
Activities:

Frame 1:
 Frame 2: The dog is looking towards the door. The dog might be waiting or anticipating something.
 Frame 3: The dog is looking at the bucket.
 Frame 4: The dog is interacting with the bucket. The dog is possibly sniffing or looking inside the bucket.
 Frame 5: The dog is looking at the bucket. The dog is interacting with the contents of the bucket.
 Frame 6: The dog is looking towards the right side. It is unclear if the dog is interacting with the yellow object or just standing next to it. Therefore, the activity is listed as "Interacting with the yellow object " to account for both possibilities.
 Frame 7: The dog is standing on its hind legs. The dog is looking at the blue bowl. The dog might be trying to reach the contents of the bowl.
 Frame 8: The dog is leaning over the bowl. The dog is eating or drinking from the bowl.

Verbs:

Frame 1:
 Frame 2: Something is looking. Something might be waiting or anticipating.
 Frame 3: Something is looking at something.
 Frame 4: Something is interacting with something. Something is investigating something.
 Frame 5: Something is looking at something. Something is interacting with something.
 Frame 6: Something is looking towards something. Someone is interacting with something .
 Frame 7: Something is standing on its hind legs. Something is looking at something. Something might be trying to reach something.
 Frame 8: Something is leaning. Something is consuming.

Figure F.3: Qualitative examples of objects, activities, and verbs textual descriptions for random videos in MSRVT dataset.



Objects:

Frame 1: hand, food, potato or similar starchy vegetable, peel, background
 Frame 2: man, dark-colored top, kitchen counter, bottle, bowl, large appliance, refrigerator or freezer, handle, kitchen cabinets, artificial lighting.
 Frame 3: person, dark green t-shirt, cutting board, vegetables, red toaster, white container, countertop, bowl, spoon.
 Frame 4: man, green top, black oven, glass door, white cabinet, handle, black refrigerator, freezer compartment, shelf, cans.
 Frame 5: hand, black object, green label, kitchen counter or table.
 Frame 6: man, dark green/black t-shirt, red appliance, window, garden or outdoor area, countertop, bottle, bowl.
 Frame 7: cutting board, broccoli, knife, countertop.
 Frame 8: person, man, dark-colored shirt, kitchen counter, kitchen cabinet, microwave, kitchen appliance.

Activities:

Frame 1: A person is peeling a potato or a similar starchy vegetable.
 Frame 2:
 Frame 3: The person is cooking or preparing food.
 Frame 4:
 Frame 5: A hand is holding an object. The hand is interacting with the object.
 Frame 6: The man is looking down.
 Frame 7: Cutting broccoli with a knife.
 Frame 8: The man is looking at a piece of paper or a recipe. The man is standing in front of a kitchen appliance, suggesting he might be preparing to use it.

Verbs:

Frame 1: Someone is peeling something.
 Frame 2:
 Frame 3: Someone is cooking or preparing.
 Frame 4:
 Frame 5: Something is holding something. Something is interacting with something.
 Frame 6: Someone is looking down.
 Frame 7: Someone is cutting something.
 Frame 8: Someone is looking at something. Someone is preparing to use something.



Objects:

Frame 1: woman, stroller, stroller, black frame, canopy, grassy area, trees.
 Frame 2: woman, stroller, stroller, grassy area, trees.
 Frame 3: white background, darker line.
 Frame 4: text, subtitle, main text, font, white background.
 Frame 5: text, two lines, sans-serif font, blue text
 Frame 6: text message, light blue background, darker blue font, text: "each of your outings with baby that much easier.", text: "that much easier."
 Frame 7: text line 1, text line 2, sans-serif font, shade of blue
 Frame 8: text, "happy", "strolling", "joggermom", ".", "com"

Activities:

Frame 1: The woman is holding onto the handles of both strollers. The woman is either pushing or maneuvering the strollers.
 Frame 2: The woman is maneuvering or pushing the strollers.
 Frame 3:
 Frame 4:
 Frame 5:
 Frame 6:
 Frame 7:
 Frame 8:

Verbs:

Frame 1: Someone is holding onto something. Someone is maneuvering something.
 Frame 2: Someone is maneuvering or pushing.
 Frame 3:
 Frame 4:
 Frame 5:
 Frame 6:
 Frame 7:
 Frame 8:

Figure F.4: Qualitative examples of objects, activities, and verbs textual descriptions for random videos in MSRVT dataset.

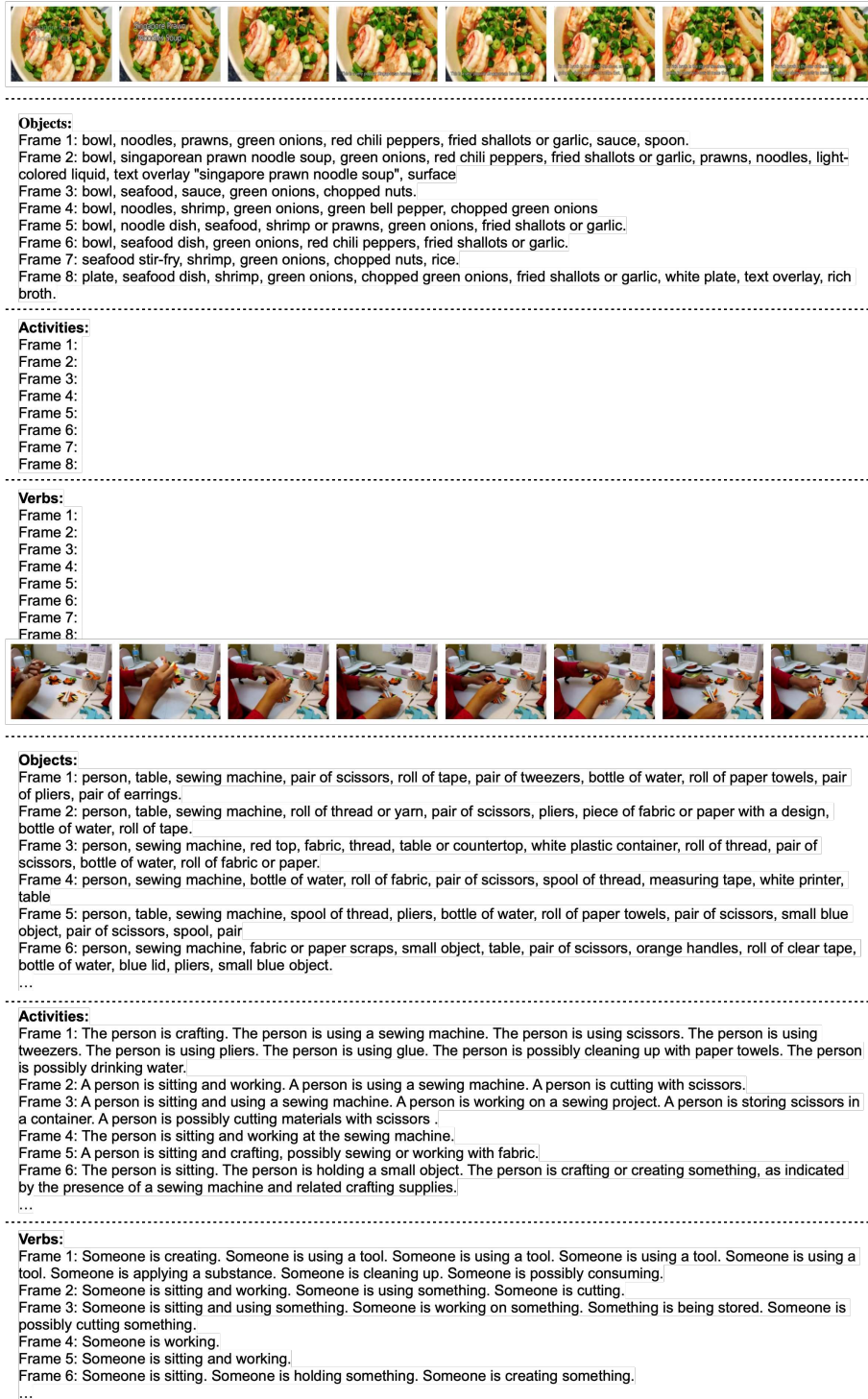


Figure F.5: Qualitative examples of objects, activities, and verbs textual descriptions for random videos in MSRVT dataset.

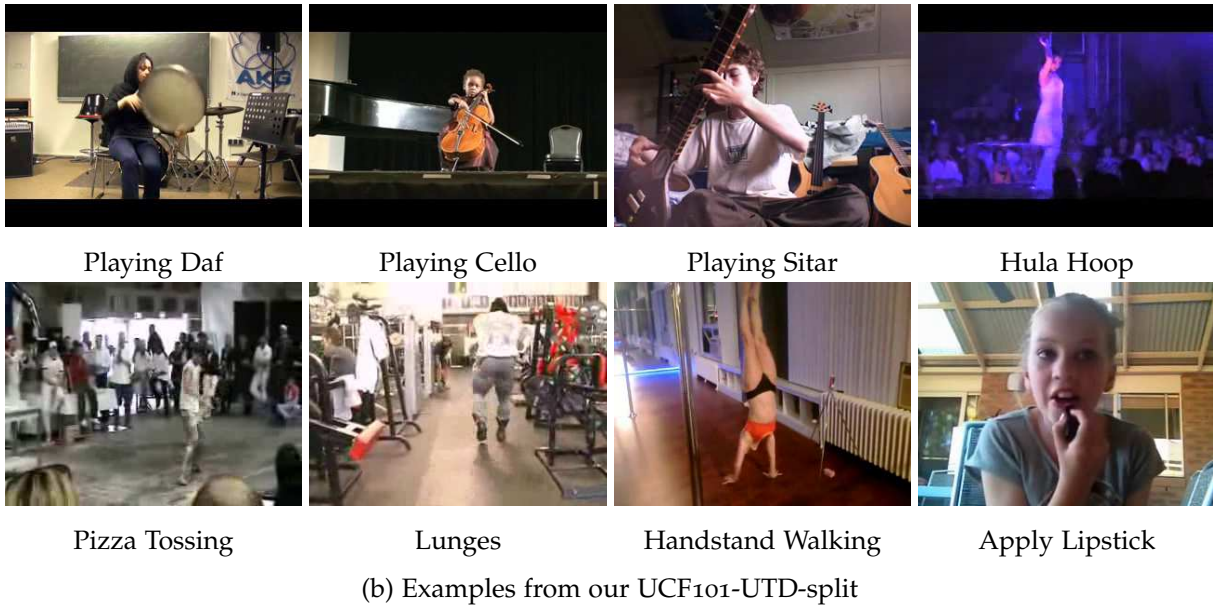
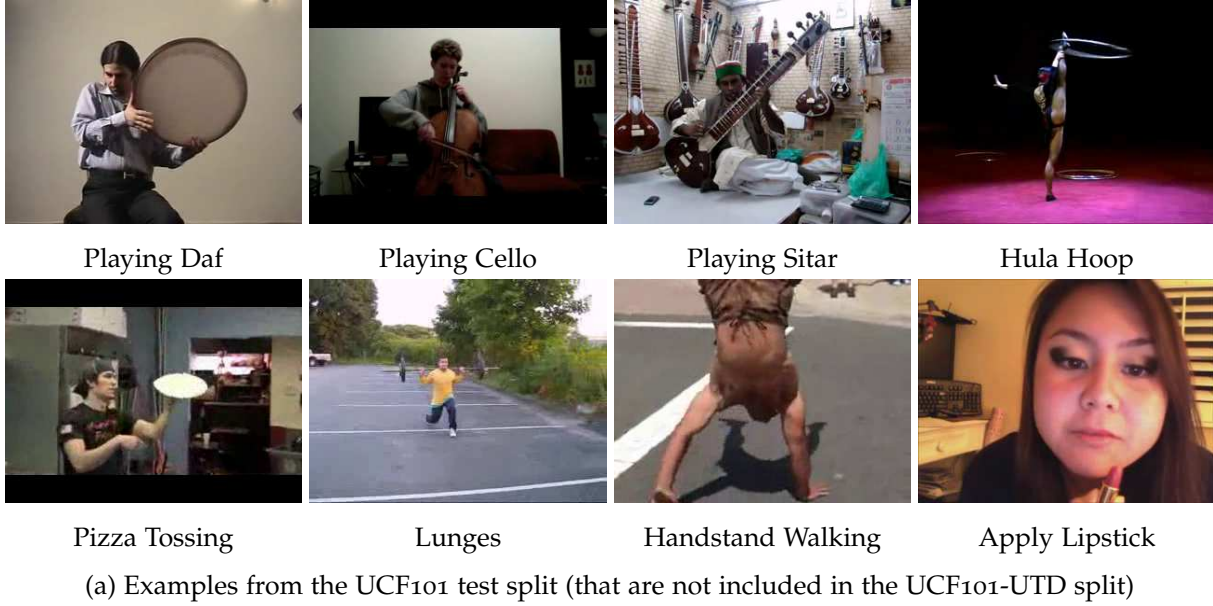


Figure F.6: Video examples with their class label from full UCF101 test set and our object-debiased UTD-split. We observe that samples from our object-debiased UTD-split require a level of video understanding beyond simple object recognition. For instance, in the case of playing musical instruments, e.g., Playing Daf or Playing Cello, the videos often include other musical instruments in the background, e.g., a piano or drums in the case of Playing Daf, alongside the primary instrument. Similarly, in the Pizza Tossing class, the pizza in the UTD-split example is hardly visible, and a video requires analysis beyond this single frame for correct class prediction.



sleeveless top



snare drum



upright piano

(a) Attribute error



platform



whiteboard or chalkboard



blue sports ball

(b) Misclassification



rows of seats



metallic base



fence

(c) Hallucination



white rails



white-capped waves



red strap

(d) Human annotation mistake – the object is visible



body of water



air



large

(e) Other

Figure F.7: Examples of manual classification of objects predicted by the VLM for the image, but not selected as visible in the image during the user study. We consider five categories: attribute error, misclassification, hallucination, human annotation mistake (the object is visible), and other.

Task	Textual Description	<instruction>
retrieval	objects o_n (seq-of-frames setup)	Given lists of objects visible on the video frames, retrieve a short video description.
		Using lists of objects seen in video frames, retrieve a brief description of the video.
		From lists of objects present in video frames, retrieve a concise video description.
	caption (from dataset)	Given a short video description, retrieve another description of this video.
		Use a brief video description as a query to retrieve an alternative description of the same video.
classification	objects o_n (seq-of-frames setup)	Given a concise video description, retrieve another description for that video.
		Given lists of objects visible on the video frames, retrieve the activity depicted in this video.
		Using lists of objects seen in video frames, retrieve the activity captured in the video.
		From lists of objects present in video frames, retrieve the activity that the video shows.

Table F.11: Multiple instructions for robust datasets unbiasing. We prompt the text embedding models using three different prompts for captions and object textual descriptions, generating three different embeddings for each. During the unbiasing process, we exclude samples from the test sets only if the sample is correctly classified or retrieved using any combination of these embeddings.

APPENDIX – WHAT, WHEN, AND WHERE?

SELF-SUPERVISED SPATIO-TEMPORAL GROUNDING IN UNTRIMMED MULTI-ACTION VIDEOS FROM NARRATED INSTRUCTIONS

This appendix contains supplementary material related to Chapter 9, *What, when, and where? Self-Supervised Spatio-Temporal Grounding in Untrimmed Multi-Action Videos from Narrated Instructions*, of this thesis. First, we provide additional details on the methodology in Section G.1 and on the experimental setup in Section G.2; then, we present further ablation and supplementary experimental results in Section G.3; and finally, we describe the annotation process and dataset analysis in Section G.4.

Contents

G.1	Method Details	275
G.2	Experimental Setup	276
G.2.1	Baseline Details	276
G.2.2	Backbones and Training	276
G.2.3	Inference	277
G.2.4	Evaluation Metrics	278
G.3	Additional Experiments	279
G.3.1	Runtime Analysis	279
G.3.2	Single-Action Spatio-Temporal Grounding	279
G.3.3	Ablation and Decision Choices	280
G.4	GroundingYoutube Annotation	282
G.4.1	Development of the Graphical User Interface and Task Description	282
G.4.2	Quality Control	284
G.4.3	Dataset Usage for Evaluation	285

G.1 METHOD DETAILS

To acquire the optimal assignment from word features to video frames, an assignment matrix $\mathbf{Q}$ is computed from each video and ASR pair as shown in Figure 9.2(a). This cross-model optimal transport mechanism is applied to assignment $\mathbf{Q}$ from the projected cross-model similarity $\mathbf{P}$ between word tokens and each video frame, where $\mathbf{P} = g(\mathcal{S}) \otimes f(\mathcal{V})^\top \in \mathbb{R}^{K \times U}$. To compute the assignment matrix, the text and video projection layers from the global representation in Figure 9.2(c) are used to project multimodal features into a common space for feature similarity calculation. To ensure that the word-to-frame assignment contains more diversity instead of just saturated assignments to a single video frame, we add a constraint that requires label assignments to be equally distributed across various video frames representing diverse

object/action concepts. This is achieved by restricting $\mathbf{Q}_v$ to a *transportation polytope* $\mathcal{Q}_v$:

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{U \times K} \mid \mathbf{Q}\mathbf{1}_K = \frac{1}{U}\mathbf{1}_U, \mathbf{Q}^\top \mathbf{1}_U = \frac{1}{K}\mathbf{1}_K \right\}, \quad (\text{G.1})$$

which enforces the soft-assignment distribution $\mathbf{Q}$ to assign an equal marginal probability to each of the U frames instead of converging to a single frame. The vector $\mathbf{1}_U$ represents one vector with dimension $U \times 1$.

The next goal is to enforce this *transportation polytope* $\mathcal{Q}$. A solution for $\mathbf{Q}$ is now computed using the optimal transport Sinkhorn-Knopp algorithm [CMM⁺20, Cut13] as shown in Figure 9.2(b). The Sinkhorn-Knopp algorithm also normalizes the distribution of $\mathbf{P}$ as:

$$\mathbf{Q} = \text{Diag}(\mathbf{f}\mathbf{f}) \exp \left(\frac{\mathbf{P}}{\varepsilon} \right) \text{Diag}(\mathbf{f}\mathbf{i}), \quad (\text{G.2})$$

where $\mathbf{f}\mathbf{f}$ and $\mathbf{f}\mathbf{i}$ are scaling vectors that restrict $\mathbf{Q}$ to have a uniform distribution across region assignment. ε is a parameter that controls the smoothness of the mapping [CMM⁺20].

The T frames are then selected by the corresponding assignment $\mathbf{Q}$ from the frames with top T aggregated similarity sum over each word for further training. Note that the selection part $\mathbf{P}$ is from a trainable projection. While acquiring a better word-to-region projection during training, we hypothesize that the frame selection also benefits. The respective frame selection strategy is evaluated in Table 9.5.

G.2 EXPERIMENTAL SETUP

G.2.1 Baseline Details

MIL-NCE [MAS⁺20], which utilizes S3D [XSH⁺18] and word2vec [MCCD13] to project two modalities into a common space, is chosen as the standard baseline for this task; CoMMA [TPS⁺21], the best-performing model for spatial representations in self-supervised learning (we denote CoMMA[†] to represent the model that uses weights shared by the author¹); CLIP [RKH⁺21b], an image-text model trained with transformer architecture, is further applied as the backbone and trained with [TPS⁺21] to construct CoMMA[†]; GLIP [LZZ⁺22] and RegionCLIP [ZYZ⁺22], state-of-the-art image-text grounding models that combine large-scale image caption pretraining and object detection finetuning, which we consider weakly supervised as the bounding box proposal network was trained on other human-annotated data. We further construct a strong baseline out of the best methods for temporal and spatial localization, MIL-NCE+RegionCLIP, where we use MIL-NCE for temporal localization and RegionCLIP for spatial grounding following the inference pipeline of Figure 9.3 without additional training.

G.2.2 Backbones and Training

We evaluate the proposed method on backbones, CLIP [RKH⁺21b] and S3D-word2vec [MAS⁺20]. We described the detailed setup as well as the training in the following.

CLIP models. For both the visual and text backbone, we use the pretrained weights from CLIP [RKH⁺21b] with transformer ViT-B/32 and fix the encoder. Both the visual and text encoder has a final embedding size of 512. We apply them to video segments with 12-28 seconds,

¹We thank the authors for providing code and weights.

processing 1 frame per second. An evaluation of how many frames to process (identical to the number of seconds) is shown in Table G.3. We sampled the video with 5 fps. It shows the best results when we start with 80 possible frames U (as described in Section 9.3.2), from which $T = 16$ frames are selected for training. Ablation of the number of frames T used for training is shown in Table G.4. We used a batch size of $B = 64$ video clips.

S3D-word2vec models. For the video backbone, we follow [TPS⁺21] and use S3D initialized by MIL-NCE on HowTo100M [MAS⁺20] at the rate of 5 frames per second and fix the video encoder. The global video clip features were max-pooled over time and projected into embeddings of dimension 512. We used the mean-pooled S3D spatio-temporal features to represent the global representation of the video following the S3D architecture [XSH⁺18]. For the text feature, we follow [MZA⁺19] using a GoogleNews pre-trained word2vec model [MCCD13] and max-pooling over words in a given sentence to acquire the text global feature. We follow [MAS⁺20] to use the max-pooled word embedding to represent the sentence (global representation) since there is no [CLS] token. Also, the sentence feature is used for the query word selection instead of the [CLS] token. We use a batch size of $B = 96$ video clips.

Training. For the training of both backbone settings, we use an Adam optimizer [KB15b] with a learning rate of $1e-4$. In the setting of finetuning CLIP, we set a learning rate of $1e-7$ for the CLIP backbone. The model is trained for 10 epochs on 4 V100 GPUs, which takes about two days.

G.2.3 Inference

Inference for the proposed model and CoMMA. For inference in the case of temporal grounding, as shown in Figure 9.3(a), we first normalize the global feature for video and text. We used a (temporal) threshold $\theta = 0.5$ to separate detections from the background. In spatial grounding, we acquire an attention heatmap using the attention rollout [AZ20] described in Section 9.3.5. We set a spatial threshold $\tau = 0.01$ to create the mask, as shown in Figure 9.3(b). The choice of this spatial threshold is evaluated in Table G.6.

GLIP, RegionCLIP baseline inference. In spatial grounding, we are given a text query and need to localize it in the frame. GLIP and RegionCLIP predict multiple bounding boxes corresponding to the text query. We select the predicted bounding box with the highest confidence score as the prediction result. We use the center point of the predicted bounding box for the pointing game evaluation as the model prediction. For mAP evaluation, we use the predicted bounding box to compute IoU with the ground truth bounding box. In spatio-temporal grounding, we input all possible action description labels as candidates similar to Figure 9.3(a). We pick the class with the highest confidence score as the predicted label. If the model made no prediction, we would predict it as “background”. The spatial inference is the same as the spatial grounding setting.

TubeDETR, STCAT baseline inference. TubeDETR and STCAT are spatio-temporal grounding models trained to predict a single spatio-temporal tube per video. In both cases, TubeDETR and STCAT, we use models trained on the Vid-STG dataset with 448x448 resolution and evaluate them for the task of spatial grounding. Since this dataset contains mostly short videos (<30 sec), we observed that both methods will also only predict a trajectory tube in this temporal range (<30 sec), no matter how long the input video is. To allow us to apply them to longer videos (>30 sec), we split the longer videos based on sliding windows of 5-sec for better performance.

MIL-NCE, CLIP baseline inference. Both models are trained based on global representations for both input modalities, videos/images and text. We can, therefore, directly compute a sentence-to-video-frame similarity to perform the temporal grounding for Figure 9.3(a),

following the same process as the proposed method for temporal grounding. For spatial grounding, we compute sentence-to-region feature similarity. Both visual backbones produce a 7x7 grid feature. We normalize the sentence and region features, then select a spatial threshold $\tau = 0.5$ to create the mask for the *mAP* evaluation.

G.2.4 Evaluation Metrics

(i) **Spatio-temporal grounding in untrimmed video** is evaluated on our annotated GroundingYoutube dataset. We combined the spatial and temporal grounding evaluation as before [KIRG19, AKB⁺19] to form the spatio-temporal evaluation. The entire video and the respective pool of action instructions were provided. The model needs to localize each action step in temporal (start-time/end-time) and spatial (location in the video) as described in Figure 9.3. We evaluate in two metrics: **IoU+Pointing game** combines the evaluation setting from the spatial grounding [AKB⁺19] and temporal grounding [KIRG19] metrics. For each video frame, the prediction is correct when the model predicts the correct action for the frame. Also, given the predicted action as a query, the maximum point of the heatmap aims to lie within the desired bounding box. We then compute the Intersection over Union (IoU) over all the predictions with the GT to acquire the final score. We also compute **video mAP** following previous evaluation [GSR⁺18], where we set IoU threshold between GT and predicted spatio-temporal tubes. A prediction is correct when it surpasses the IoU threshold. We then compute the mAP over all classes. We form a 3D prediction mask following Figure 9.3 and compute IoU between our 3D heatmap and 3D tube.

(ii) **Spatial grounding** is given a text query description to localize the corresponding region in the trimmed video. We use GroundingYoutube, Youcook-Interaction, V-HICO, and Daly for evaluation. This task is evaluated using the **pointing game accuracy**. Given the query text and video, we compute the attention heatmap on the video as described in Figure 9.3(b). If the highest attention similarity score lies in the ground truth bounding box, the result counts as a “hit” and counts as “miss” otherwise. The final accuracy is calculated as a ratio between hits to the total number of predictions $\frac{\# \text{ hits}}{\# \text{ hits} + \# \text{ misses}}$. We report the mean average precision (**mAP**) following the settings from V-HICO [LDT⁺21]. Given a human-object category as the text query, we aim to localize the spatial location in the video frame. The predicted location is correct if their Intersection over-Union (IoU) with ground truth bounding boxes is larger than 0.3. Since we do not use any bounding box proposal tools or supervision, we create an attention heatmap as described in Figure 9.3(b) to create a mask for IoU computation. We follow [LDT⁺21] and compute the mAP over all verb-object classes.

(iii) **Temporal grounding** provides videos with the respective actions and their ordering, including the background. The goal is to find the correct frame-wise segmentation of the video. We follow the inference procedure in [KIRG19] to compute the alignment given our similarity input matrix. The task is evaluated by intersection over detection (IoD), defined as $\frac{G \cap D}{D}$ the ratio between the intersection of ground-truth action G and prediction D to prediction D , and the Jaccard index, which is an (IoU) given as $\frac{G \cap D}{G \cup D}$.

Method	Backbone	DataSet	Supervision	Modality	GroundingYoutube						
					IoU+Point	mAP					
						0.1	0.2	0.3	0.4	0.5	0.1:0.5
CoMMA* [TPS ⁺ 21]	S3D-word2vec	HT250K	Self	VT	1.10	7.46	5.84	4.20	2.65	1.53	4.93
MIL-NCE [MAS ⁺ 20]	S3D-word2vec	HT100M	Self	VT	12.41	45.91	32.33	15.35	3.70	2.56	19.54
Ours	S3D-word2vec	HT200K	Self	VT	19.46	51.95	40.31	26.81	16.27	7.81	28.63
CoMMA+ [TPS ⁺ 21]	CLIP	HT200M	Self	VT	2.64	8.94	6.89	5.47	4.18	2.67	5.63
CLIP [RKH ⁺ 21b]	CLIP	HT200K	Self	IT	11.34	43.28	30.64	11.20	3.10	1.94	18.03
RegionCLIP [ZYZ ⁺ 22]	ResNet-101	CC3M	Weak	IT	17.42	51.86	40.23	26.10	15.23	7.29	28.14
GLIP [LZZ ⁺ 22]	Swin-L	Cap24M	Weak	IT	18.15	52.61	41.83	26.93	17.23	8.46	29.41
Ours	CLIP	HT200K	Self	VT	20.81	53.24	42.96	29.17	20.36	11.84	31.51
TubeDETR [YMS ⁺ 22]	MDETR	Vid-STG	Full	VT	26.43	63.47	50.95	38.23	28.31	19.34	40.06
STCAT [JyIYM22]	ResNet-101	Vid-STG	Full	VT	27.84	64.96	52.13	40.61	30.49	20.55	41.75

Table G.1: **Single-action spatio-temporal grounding in short videos.** We compare spatio-temporal grounding approaches based on single phrase grounding. To this end, we construct a clip-level evaluation based on the action segments of GroundingYouTube, where each action segment varies from 9 sec to 60 sec. We append video segments before and after the annotated action with the same time length of the action step to form the final video clip. This allows us to directly compare with supervised spatio-temporal grounding methods [YMS⁺22, JyIYM22].

G.3 ADDITIONAL EXPERIMENTS

G.3.1 Runtime Analysis

We analyze the computational costs of sampling and loss. We sample 16-second videos at a frame rate of 5 FPS (80 frames in total). We report the execution time for a single batch (batch size = 64) averaged over 100 batches. For the *frame sampling strategy*: (1) Random select 8 frames: 1.48s. (2) Optimal transport based selection of 8 frames out of 64: 1.54s. (3) Entire 64 frames: 1.74s. The execution time of our method is close to traditional random sampling while capturing diverse visual concepts, which improves the training process. For the *global and local* components: (1) Global loss only: 1.1s. (2) Local loss only: 1.52s. (3) Both losses: 1.54s. Computation of the local loss is more time-consuming than the global loss due to its requirement for features with finer granularity.

G.3.2 Single-Action Spatio-Temporal Grounding

Current spatio-temporal detection and grounding datasets [JLZ⁺14, GSR⁺18] usually aim to discriminate a single given action class from the background class in a short clip. This differs from our setup of spatio-temporal grounding in untrimmed videos, which usually comprises a set of phrases that need to be detected in a 3-5 min long video. To allow an evaluation of spatio-temporal grounding approaches based on single phrase grounding, we construct a clip-level evaluation where the clip varies from 9 sec to 60 sec. Given an action step, we append the video segments before and after the steps with the same time length of the action step to form the final video clip. This results in 2,895 clips for the spatio-temporal clip grounding evaluation. For each clip, the temporal action intervals occupy 33% of corresponding videos, which demonstrates the difficulty of the setting. In this setting, instead of selecting the possible action step from a pool, the ground truth action step was given as the text query for spatio-temporal grounding. This allows us to directly compare with supervised spatio-temporal

grounding methods [YMS⁺22, JyLYM22] as described in Section 9.5.4. As shown in Table G.1, we observe that the baseline GLIP models achieve a much better performance compared to Table 9.2. This is due to the fact that this setting does not require the model to select the text query from the pool, which the GLIP model was not trained to do. Moreover, we find that weakly supervised methods, GLIP and RegionCLIP, show only limited ability to differentiate the queried action from the background, which leads the model to ground the text query in most of the frames. However, both demonstrate powerful localization ability in foreground action segments, which results in a decent performance. The fully-supervised trained models (TubeDETR, STCAT) achieved a balance in localizing temporally and spatially, resulting in the best performance on this task.

Attention Architecture	GroundingYT Spatio-temporal	MiningYT Temporal	YouCook-Inter. Spatial
Self+Cross	15.4	18.7	54.1
Cross+Self	15.9	18.9	54.5
Cross+Cross	16.5	19.3	56.2
Cross+Self+Cross	17.1	19.9	57.1

Table G.2: Ablation on different attention architecture

# of frames	60	80	100	120	140
GYT (Spatio-temporal)	16.4	17.1	17.0	16.8	16.1
YC-Inter (Spatial)	56.3	57.1	56.8	56.7	55.9

Table G.3: Ablation of # of frames used for selection

Frame length	1	4	8	16	24
GYT (Spatio-temporal)	5.2	9.5	16.1	17.1	16.5
YC-Inter (Spatial)	31.1	48.2	55.5	57.1	56.1

Table G.4: Effect of # video frames used for training

G.3.3 Ablation and Decision Choices

We performed additional ablation studies using the CLIP backbone without finetuning.

Attention architecture. We tested different architectures by stacking the self-attention or cross-attention block in the model to calculate contextualized local representations, as shown in Figure 9.2(d). As shown in Table G.2, we found that the standard multimodal transformer architecture (self+cross) to have the worst performance. Using two cross-attention blocks was beneficial in incorporating more cross-modal interaction between local features. Finally, including a self-attention layer slightly improves the final representations by encoding better single-modality representations.

Frames used for selection. As shown in Table G.3, we perform an ablation study on the number of candidates frames U used for training. We found that selecting 80 frames (16 seconds) achieves the best performance, comprising the useful video information in training

Train/test supervision	VT/VT	VAT/VT	VAT/VAT
GYT (Spatio-temporal)	16.2	16.8	17.0
YC-Inter (Spatial)	53.9	53.6	53.8

Table G.5: Effect of audio supervision in train and test

Threshold	Backbone	0.1	0.05	0.01	0.005	0.001
CoMMA*	S3D-word2vec	0.76	0.90	0.93	0.91	0.86
Ours	S3D-word2vec	15.35	15.88	16.22	16.34	16.12
CoMMA†	CLIP	0.88	0.92	0.99	0.94	0.91
Ours	CLIP	15.93	16.33	17.10	17.05	16.24

Table G.6: Threshold for attention score on GroundingYoutube $mAP@0.4$

while not including too many irrelevant concepts that diverge from the action/object in the ASR sentence.

Number of frames for training. We further evaluated the impact of different numbers of frames T used for training. As shown in Table G.4, selecting fewer frames for training significantly causes the performance to drop. We hypothesize that the model not only fail to capture the temporal dynamics with fewer frames but also loses some frames with groundable objects in the sentence while training. We also hypothesize that with a too large number of frames, more irrelevant frames might be selected during training, which decreases the performance.

Effect of audio in training and testing. Unlike text which describes a discrete concept as a target to ground, audio serves as a continuous representation that is highly relevant to the temporal information. For example, we can determine an action started when we hear a “cracking” sound. In Table G.5, we tested our model using the additional audio modality. For the audio branch, we compute log-mel spectrograms and use a DAVeNet model [HRS⁺18] initialized by MCN on HowTo100M [CRD⁺21] to extract audio features. We extend the global and local loss pairs from VT to VT, VA, and AT following [SCR⁺22]. We found when training and testing with audio, the spatio-temporal result increases the temporal performance while the spatial-only result remains the same. This validates our assumption that audio contributes more to temporal understanding. When we trained on audio and tested without audio, the performance increased over the VT model, showing that the audio serves as useful supervision for better video/text representations.

Threshold for attention mask. As shown in Figure 9.3(b), we apply a threshold to create a mask from the result of attention rollout. Note that this threshold τ is not a hyperparameter that affects the training or the model but simply serves as a means to an end to compute the mAP scores. We did not systematically optimize this threshold, but instead, Test different thresholds for attention scores for all relevant models (COMMA, ours) using the spatio-temporal grounding mAP IoU@0.4 on our GroundingYoutube dataset as shown in Table G.6. We find 0.01 to be a reasonable threshold among all models, performing best on COMMA and giving at least the second best results for the proposed model.

Mean pooling for global features We also tried mean pooling over all tokens for CLIP to replace [CLS] for the global feature. As shown in Table G.7, [CLS] outperforms mean pooling in our 3 datasets. We attribute this to the fact that [CLS] was calculated by self-attention, which will automatically select important tokens, whereas mean pooling treats all tokens with the same importance.

Loss ablation with MIL-NCE(S3D) We tested the local vs. global loss on top of S3D (initialized

	GroundingYT Spatio-temp	MiningYT Temporal	YC-Inter. Spatial
Mean-pooling	18.39	19.24	57.84
[CLS]	19.45	20.33	58.35

Table G.7: **Global representation ablation with CLIP.** [CLS] showed superior performance due to self-attention architecture.

	GroundingYT Spatio-temp	MiningYT Temporal	YC-Inter. Spatial
only Local loss	5.62	4.97	50.54
only Global loss	7.53	18.67	31.51
w/ Both loss	16.22	19.18	53.98

Table G.8: **Loss ablations with MIL-NCE (S₃D).** Results show a similar trend as the CLIP backbone used in the main part of the thesis.

by MIL-NCE, see Table G.8), showing similar behavior compared to CLIP (shown in Table 9.6 in the main part of the thesis).

G.4 GROUNDINGYOUTUBE ANNOTATION

The data annotation was divided into three phases: During *Phase I* (Appendix G.4.1, a graphical user interface (UI) and the task description were developed. In *Phase II*, the dataset was given to the annotators to generate the key points (Appendix G.4.1). In *Phase III*, a manual quality control step was performed (Appendix G.4.2).

G.4.1 Development of the Graphical User Interface and Task Description

The annotation of a large amount of data is often one of the most expensive aspects of a machine learning pipeline design, which is why the annotation time per datum should be kept as short as possible. There are two points that can be optimized, (1) the training or the task “message” for the annotators and (2) the graphical user interface by minimizing interaction times.

While tasks are usually formulated in such a way that no ambiguities arise, i.e. all possible edge cases are somehow covered, and simple words are used, in this case, we made a conscious decision to choose questions as short as possible, and that would give the annotator room for interpretation. We did this because it was hard to predict where people would actually locate actions in images. We also created a 1 min 30 sec long user training video where we demonstrate the task using exemplary keypoint annotations and explain how to use the UI.

Our annotation UI was designed with a special focus to keep it as intuitive as possible and reducing the interaction time. Our UI only provided five functionalities (set/unset a keypoint, undo the last image, image can’t be solved, and image is corrupt) which were clearly described in text buttons (see Figure G.1). Further, to reduce the cognitive load of our workers, images were presented in the form of work packages, each containing 25 images. Hence, we could ensure that completing a task would take no longer than 6 minutes.

The annotation of all 26,987 images was performed with five distinct repeats per image, resulting in 134,935 labels in total. All labels were generated by 13 professional annotators in

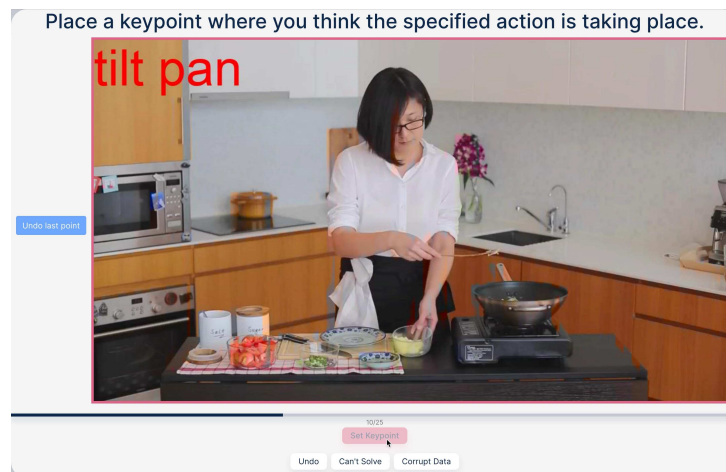


Figure G.1: A screenshot of our simplified annotation interface. On the top, the annotation task is described in simple and short words to save reading time. To make interacting with the UI as intuitive as possible, actions are limited to simple button clicks and setting the key point by clicking on the image.

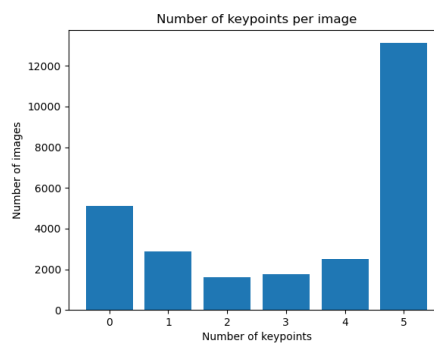


Figure G.2: **Number of keypoints per image.** It can be seen that 48% of the data has all 5 key points and 19% has not a single annotation

total, which took them 5s in average per image. However, it should be noted that the number of images where an annotator placed a keypoint differs along all the workers (see Figure G.2) and that the vast majority of all images have been answered by five annotators only. Examples are shown in Figure G.3.

During the annotation, professional annotators were given a short instruction video at the beginning and then asked to click on the center of the given action without additional instructions. They were further free to choose “can’t answer” if they could not locate the action, e.g., at the beginning and end of the clip. Thus, the number of available key points per image differs, and we choose majority voting to determine whether an action is present, resulting in new, refined temporal boundaries compared to the original annotation.

We found that the point-wise annotation resulted in roughly three distinct patterns, which depend on the captured scenario, as shown in Figure G.4. In the case of half portrait or even wider shots in Figure G.4a, annotations are highly locally centered. We further found that in some cases, the point annotation can also represent the flow of the action, e.g., pouring oil in Figure G.4b, or even split into two separate clusters in Figure G.4c.

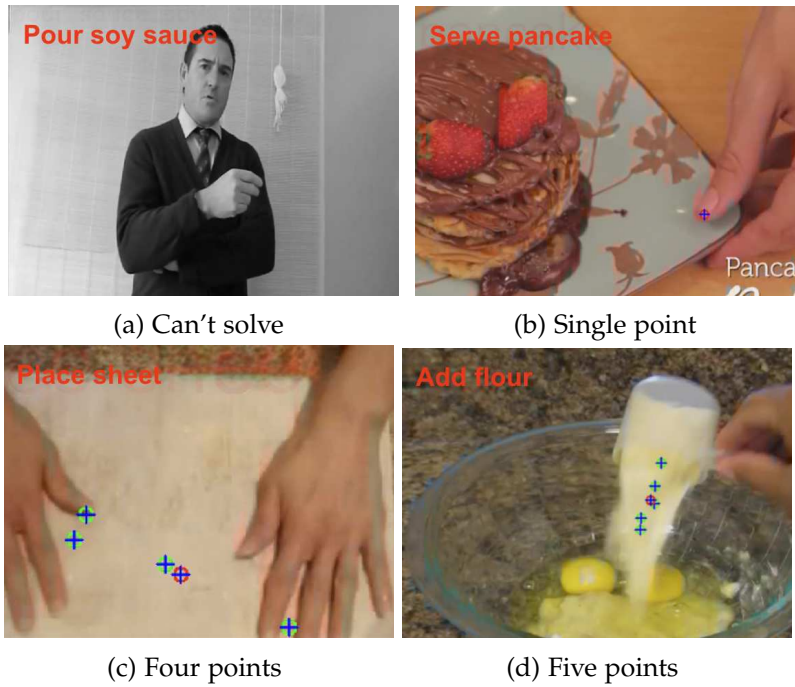


Figure G.3: **Sample annotations.** The purple point represents the center point of the annotations in the frame. 48% of the data has all 5 key points, and 19% has not had a single annotation.

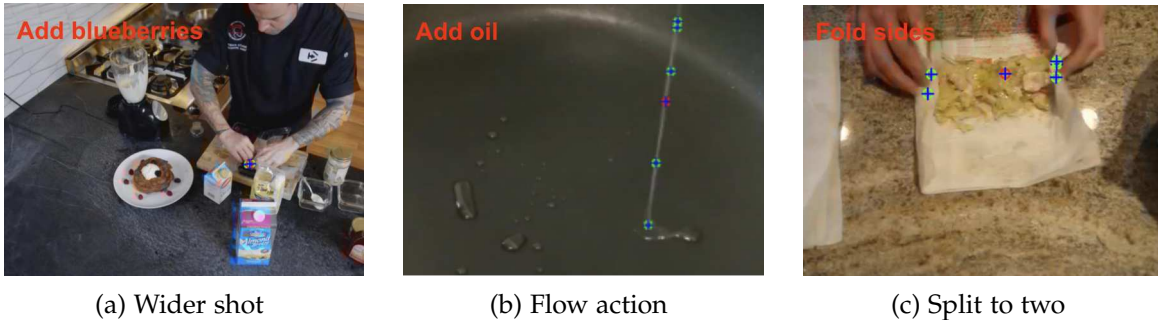


Figure G.4: **Example of keypoint annotations under different conditions.**

G.4.2 Quality Control

Since the label quality of the datasets used is a critical factor in the performance of machine learning models, we verified the correctness of a subset of our images using an experienced annotation specialist for 1,026 randomly selected frames. To evaluate the data quality, we evaluate the agreement between the annotation specialist and the annotations provided by the annotators. To this end, we considered an annotation as a false positive if three annotators or more have set a key point, although no action can be seen in the image, and as a false negative if three annotators or more have not set a key point, even though an action can be seen in the image. The entire sample was assessed using these criteria, with the specialist disagreeing with the annotators in only a total of $1.1\% \pm 3\%$ (FP: $0.7\% \pm 3\%$, FN: $0.4\% \pm 3\%$). We also found that annotations significantly diverted in terms of spread. Namely, wider shots tend to be highly centered, whereas zooming in together with the usage of larger objects such as a pan or a spatula results in more widespread key points. We also analyzed how often those cases

Distribution Type	$mAP@0.4$
Widespread actions	18.34
Saturated actions	15.96
Total	17.10

Table G.9: Performance on the annotation distribution types of widespread v.s. saturated.

occur and found that 14.0% of the selected frames show a widespread pattern.

Sample size calculation To this end, we first needed a representative subset of N_S images of our data. We calculated the required sample size based on the following two formulas:

$$N_0 = \frac{z^2}{\epsilon^2} \cdot p \cdot (1 - p) \quad (G.3)$$

where α is the confidence interval, p the expected probability of the appearance of a quality aspect (e.g., widespread answers), ϵ is the accepted error margin, and $Q(\alpha)$ is the percent point function of a normal distribution and $z = Q(1 - \frac{\alpha}{2})$.

As N_0 would be the required sample size for an infinitely large population, we applied the finite population factor that results from sampling without replacement from a finite population.

$$N_S = \frac{(N_0 \cdot N)}{N_0 + (N - 1)} \quad (G.4)$$

where N is the total number of images.

We set $\alpha = 95\%$, $\epsilon = 3\%$, and our sample size of $N = 26,987$. As the probability of the quality aspect is unknown, we set $p = 50\%$, which resulted in 1,026 being checked for quality control.

G.4.3 Dataset Usage for Evaluation

How to derive bounding box? We derive bounding boxes by adding a constant distance ($0.15 \times \text{frame_width}$ in width, $0.15 \times \text{frame_height}$ in height) to the boundaries of a union of all annotated points (as shown in Figure 9.4). Since point annotations may be scattered in the image or, conversely, gathered around a point, the output bounding boxes will vary in size over time and for different actions. We manually check the auto-generated bounding boxes and adjust the bounding box when needed.

Performance on widespread and saturated action. We evaluate the performance of different action distributions using the spatio-temporal grounding mAP IoU@0.4 setting. We define widespread actions to have an area larger than a certain threshold A . Here, we set $A = 60,000$ pixels. As shown in Table G.9, the performance of the widespread actions was higher since it had a higher tolerance of spatial localization error.

APPENDIX – VIDEOGEM: TRAINING-FREE ACTION GROUNDING IN VIDEOS

This appendix contains supplementary material related to Chapter 10, *VideoGEM: Training-free Action Grounding in Videos*, of this thesis. First, we show a comparison between GEM and our proposed weighting mechanism in Section H.1. Next, we present additional experimental evaluation in Section H.2 and qualitative results in Section H.3. Finally, we discuss limitations in Section H.4.

Contents

H.1	Weighted GEM vs. GEM	287
H.2	Additional Experimental Evaluation	288
H.2.1	Additional Backbone	288
H.2.2	Video Input for Image-Language Models	289
H.2.3	Ablation for Static and Dynamic Weights	290
H.2.4	Ablation for Weight Coefficients	291
H.2.5	Comparison to Multimodal LLMs	291
H.2.6	Ablation for CLIP Prompting	293
H.3	Qualitative Analysis	293
H.4	Limitations	296

H.1 WEIGHTED GEM vs. GEM

VideoGEM contains two new concepts that we introduced in the main part of the thesis: prompt decomposition and a weighting mechanism for GEM. While prompt decomposition has already been illustrated in Figures 10.1 and 10.2, we also want to focus on the difference between GEM and our proposed weighting mechanism for GEM (weighted GEM) in Figure H.1. The proposed weighting mechanism is illustrated on the left. Static weights and dynamic weights can be applied independent of each other. While static weights can be set heuristically or via hyperparameter search based on the general pipeline performance, dynamic weights are adapting to the importance of the different transformer layers individually and with respect to each prompt as described in Section 10.3.3. Standard GEM does not use any weights, which equals to always using 1—weights in our formulation. This results in one weight for the initial self attention output that is the first input into GEM as well as one weight for the output of each self-self attention block. A layer output is multiplied by its corresponding weight, producing the final output as the sum of weighted outputs.

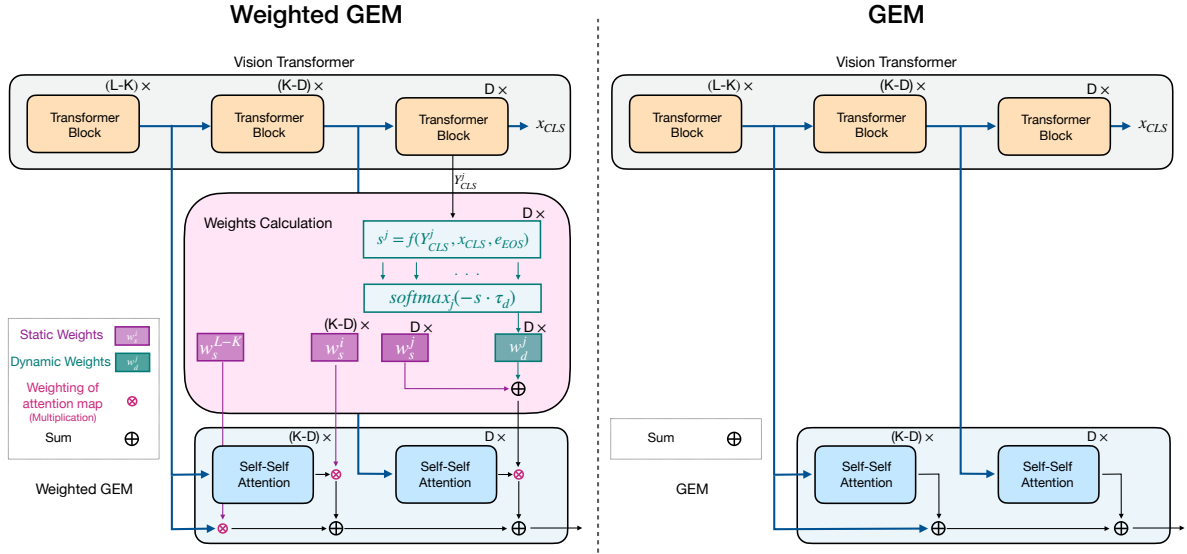


Figure H.1: **Comparison of GEM and our proposed weighting mechanism.** The proposed weighting mechanism is illustrated on the left. Static weights and dynamic weights can be applied independent of each other. While static weights can be set heuristically or via hyperparameter search based on the general pipeline performance, dynamic weights are adapting to the importance of the different transformer layers individually and with respect to each prompt as described in Section 10.3.3. Standard GEM does not use any weights, which equals to always using 1 – weights in our formulation.

H.2 ADDITIONAL EXPERIMENTAL EVALUATION

H.2.1 Additional Backbone

We extend the evaluation of our VideoGEM to another backbone. Namely, we consider the image-text BLIP [LLXH22b] model finetuned on the instructional video-text HowToCaption dataset [SKH⁺24] (Table H.1). The HowToCaption dataset is based on the HowTo100M [MZA⁺19] dataset and provides captions for instructional videos. We apply both, the original GEM as well as the proposed VideoGEM to this backbone and evaluate it on the V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT) datasets. Table H.1 shows that also with BLIP as a backbone, VideoGEM outperforms GEM by more than 10% on average, outperforming GEM individually on each dataset. Since our proposed prompt decomposition method relies on good predictions for the individual verb, object, and action prompts we assume that the proposed method benefits more from a finetuned backbone with improved individual predictions via prompt decomposition than the original GEM. Moreover, Tables H.3 to H.5 show that BLIP also largely benefits from dynamic and static weights.

Setting	VH	Daly	YC	gYT	avg
GEM	67.79	69.00	34.77	37.97	52.38
VideoGEM	77.20	72.04	51.57	53.83	63.66

Table H.1: **Experimental evaluation using the BLIP backbone fine-tuned on the video-text HowToCaption dataset.** We report the model’s performance on the V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT) datasets.

Model	Set	Data	VH	Daly	YC	gYT	avg
OpenCLIP	base	img	68.28	74.05	56.87	32.91	58.03
		vid	65.20	70.68	48.67	28.15	53.18
	ours	img	76.42	80.32	60.05	45.33	65.53
		vid	74.26	76.55	52.93	42.10	61.46
CLIP	base	img	67.79	78.52	50.08	36.92	58.33
		vid	67.55	76.31	46.10	34.82	56.20
	ours	img	76.90	84.53	52.57	47.46	65.37
		vid	75.15	82.68	50.36	45.79	63.50

Table H.2: **Experimental evaluation of extending image-text backbones to video input.** CLIP and OpenCLIP are extended to video input in a training-free manner. Their accuracy with image and video data as input is compared for VideoGEM (ours) and standard GEM (base) on the V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT) datasets.

H.2.2 Video Input for Image-Language Models

Since image-language models, such as CLIP and OpenCLIP, are limited to processing single images as input, we conduct additional experiments where we extend these models to handle video input.

Setup. Let $F = \{f_1, \dots, f_T\}$ be a video with T frames. For an input frame f_i , the first layer l_0 of a backbone encodes the input image into N patch tokens, before they are jointly processed through the transformer layers. In order to utilize video data we apply the first layer l_0 individually to all frames $f_i, 1 \leq i \leq T$ resulting in $T * N$ patch tokens. Since the transformer layers are independent of the number of patch tokens, the $T * N$ patch tokens are processed by the transformer layers without any architectural change. For a given input sequence, the patch tokens from all frames are concatenated into a single sequence $X \in \mathcal{R}^{(T*N) \times d}$, where d is the embedding dimension and further processed by GEM as described in Section 10.3.2.

Results. We evaluate the performance of GEM and VideoGEM with CLIP and OpenCLIP as backbones for video input with $T = 8$ frames compared to image input in Table H.2. The frame sampling for video input is the same as for ViCLIP in previous experiments according to Section 10.4.2. Independent of the backbone or the setting (GEM, or VideoGEM) video input decreases the accuracy compared to image input. We attribute this to the fact, that the backbones are not trained to handle video input and therefore are not capable of utilizing the spatial and temporal relationships that are inserted into the input data by using videos, but are distorted by the different input instead. This assumption is supported by the results for ViCLIP in Table 10.5. ViCLIP has a similar architecture to CLIP besides its first embedding layer which embeds a video input into patch tokens. ViCLIP is specifically trained with video

Model	Data	Set	VH	Daly	YC	gYT	avg
ViCLIP	vid	base	63.33	75.62	33.84	29.40	50.55
		dyn	63.27	75.82	34.00	29.62	50.68
		stat	64.66	76.20	36.17	30.96	52.00
		s+d	64.72	76.21	36.54	31.26	52.18
ViCLIP	img	base	64.17	75.87	32.64	28.40	50.27
		dyn	64.17	76.03	32.60	28.62	50.36
		stat	65.44	76.35	34.16	29.67	51.41
		s+d	65.38	76.32	34.24	30.00	51.49
CLIP	img	base	69.90	79.71	29.30	23.29	50.55
		dyn	70.14	79.79	29.14	22.57	50.41
		stat	72.32	82.06	30.43	26.44	52.81
		s+d	72.80	82.48	29.74	25.69	52.68
OpenCLIP	img	base	70.87	78.65	41.16	17.75	52.11
		dyn	70.57	78.42	41.72	18.43	52.29
		stat	69.36	80.69	43.09	20.59	53.43
		s+d	69.60	80.24	43.45	21.60	53.72
BLIP	img	base	63.63	64.75	33.00	31.18	48.17
		dyn	64.66	65.45	33.20	31.41	48.68
		stat	68.15	68.23	44.86	40.41	55.41
		s+d	69.00	67.91	45.54	40.49	55.74

Table H.3: **Ablation study on the effect of weights for verb prompts.** Accuracy for verb prompts for vanilla GEM (base), dynamic weights for the last three layers (dyn), static weights (stat), and the proposed combination of static and dynamic weights (s+d), evaluated on the V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT) datasets.

input and its accuracy benefits from using videos as input as shown in Table 10.5, suggesting that video-specific pretraining is needed to leverage the spatial and temporal relationships in videos.

H.2.3 Ablation for Static and Dynamic Weights

In this section, we extend our analysis of the effects of static, dynamic, and combined weights (elaborating on the experimental evaluation in Section 10.4.4). We investigate their effects separately for verb prompts (Table H.3), object prompts (Table H.4), and action prompts (Table H.5). Note, that compared to Table 10.2 in the main part of the thesis, Tables H.3 to H.5 don't apply prompt decomposition but only the weighting strategies. We include ViCLIP on video and image data, as well as CLIP, OpenCLIP, and BLIP which is finetuned on HowToCaption, as backbones. First, we observe that independent of the prompt (verb, object, or action) averaged over datasets and backbones, dynamic weights and static weights improve over using no weights. The combination of static and dynamic weights further improves over both, static weights or dynamic weights on their own, showing the capabilities of static and dynamic weights as well as their additive effects. Moreover, OpenCLIP and BLIP improve around 0.5 – 1% with dynamic weights across different prompts, while the accuracy of ViCLIP and CLIP only shows minor changes. This confirms the assumption that the benefit of dynamic weights highly depends on the backbone.

Model	Data	Set	VH	Daly	YC	gYT	avg
ViCLIP	vid	base	62.48	66.44	53.70	45.72	57.09
		dyn	62.00	66.53	53.82	45.75	57.03
		stat	62.24	68.73	54.86	46.56	58.10
		s+d	62.30	68.71	54.94	46.78	58.18
ViCLIP	img	base	62.06	66.83	53.86	44.39	56.81
		dyn	62.18	66.92	53.98	44.63	56.93
		stat	60.92	68.59	54.14	45.03	57.17
		s+d	61.10	68.62	54.30	45.17	57.30
CLIP	img	base	65.74	76.27	46.50	36.24	56.19
		dyn	66.47	76.78	46.02	35.70	56.24
		stat	65.68	78.70	47.35	38.06	57.45
		s+d	65.56	79.45	47.19	37.52	57.43
OpenCLIP	img	base	67.55	76.43	49.68	28.35	55.50
		dyn	68.52	75.67	50.76	30.91	56.47
		stat	64.90	77.86	50.96	32.03	56.44
		s+d	65.44	77.00	51.21	34.20	56.96
BLIP	img	base	59.95	59.85	32.68	34.17	46.66
		dyn	60.62	59.59	33.76	35.57	47.39
		stat	62.48	58.09	45.34	40.27	51.55
		s+d	62.06	57.94	46.38	42.15	52.13

Table H.4: **Ablation study on the effect of weights for object prompts.** Accuracy for object prompts for vanilla GEM (base), dynamic weights for the last three layers (dyn), static weights (stat), and the proposed combination of static and dynamic weights (s+d), evaluated on the V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT) datasets.

H.2.4 Ablation for Weight Coefficients

For prompt decomposition the same weights of $w_{verb} = 0.2, w_{obj} = 0.2, w_{act} = 0.6$ are applied in all experiments. To analyze the importance of the weighting scheme we compare 4 different weighting schemes in Table H.6. One that prioritizes objects (W_1), one that prioritizes verbs (W_2), one that treats all three prompts equally (W_3), and our original weighting scheme (org). We observe similar accuracy for the different weighting schemes, while our original weights perform the best.

H.2.5 Comparison to Multimodal LLMs

We extend our comparison to State-of-the-art methods to multimodal large language models, namely Qwen-VL [BBY⁺23]. Note that Qwen is not directly comparable, as it utilizes location information during training and differs significantly in both the number of parameters and the amount of training data. We provide a comparison of Qwen-VL and VideoGEM in Table 10.4. We use "Generate grounding for " as a prompt template for Qwen-VL. We use a different prompt for Qwen-VL compared to VideoGEM to ensure the highest rate of bounding box predictions for Qwen-VL. The center of the predicted bounding box is the prediction for Qwen-VL. The results indicate that our training-free method VideoGEM outperforms Qwen-VL on average.

Model	Data	Set	VH	Daly	YC	gYT	avg
ViCLIP	vid	base	65.08	73.75	53.62	51.28	60.93
		dyn	64.84	73.81	53.62	51.25	60.88
		stat	66.53	74.23	52.97	51.93	61.42
		s+d	65.68	74.17	52.97	51.99	61.20
ViCLIP	img	base	65.20	74.00	52.17	48.80	60.04
		dyn	64.66	74.04	52.37	48.91	60.00
		stat	65.86	74.62	52.65	49.51	60.66
		s+d	65.80	74.50	52.85	49.66	60.70
CLIP	img	base	67.79	78.52	50.08	36.92	58.33
		dyn	67.43	78.47	49.68	36.83	58.10
		stat	68.76	80.91	51.37	39.65	60.17
		s+d	68.94	80.53	50.84	39.31	59.91
OpenCLIP	img	base	68.28	74.05	56.87	32.91	58.03
		dyn	68.64	74.24	56.47	35.58	58.73
		stat	66.10	75.50	58.68	36.18	59.12
		s+d	66.77	74.87	57.36	38.38	59.35
BLIP	img	base	67.79	69.00	34.77	37.97	52.38
		dyn	69.00	68.86	37.02	39.47	53.59
		stat	69.30	71.86	46.26	45.52	58.24
		s+d	70.02	70.68	47.99	46.53	58.81

Table H.5: **Ablation study on the effect of weights for action prompts.** Accuracy for action prompts with vanilla GEM (base), dynamic weights for the last three layers (dyn), static weights (stat), and the proposed combination of static and dynamic weights (s+d), evaluated on the V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT) datasets.

Setting	VH	Daly	YC	gYT	avg
W1	73.52	77.27	56.55	56.43	65.94
W2	75.57	78.38	52.37	54.99	65.33
W3	77.80	78.74	52.33	54.98	65.96
org	75.75	78.25	55.10	57.21	66.58

Table H.6: **Influence of different weighting schemes.** VideoGEM with prompt decomposition and combined weights is evaluated with different weighting schemes: $w_{verb} = 0.1, w_{obj} = 0.3, w_{act} = 0.6$ (W1), $w_{verb} = 0.3, w_{obj} = 0.1, w_{act} = 0.6$ (W2), $w_{verb} = \frac{1}{3}, w_{obj} = \frac{1}{3}, w_{act} = \frac{1}{3}$ (W3), $w_{verb} = 0.2, w_{obj} = 0.2, w_{act} = 0.6$ (org). Where W1 prioritizes objects, W2 prioritizes verbs, W3 treats all prompts equally, and org are the original weights employed in the main part of the thesis. The weighting schemes are evaluated with ViCLIP on video input on V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT).

Model	VH	Daly	YC	gYT	avg
Qwen-VL	84.20	63.05	28.49	57.69	58.36
VideoGEM	75.74	78.25	55.10	57.21	66.58

Table H.7: **Comparison to multimodal LLMs.** VideoGEM is compared to a multimodal large language model, namely Qwen-VL [BBY⁺23]. For Qwen-VL the prompt template "Generate grounding for " is applied. The center of the predicted bounding box is the prediction of Qwen-VL. VideoGEM is applied with ViCLIP as a backbone on video input. Both models are evaluated on V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT).

Setting	VH	Daly	YC	gYT	avg
avg. txt.	63.33	63.10	51.09	54.37	57.97
avg. heat.	63.33	63.04	51.09	54.41	57.97
VideoGEM	75.75	78.25	55.10	57.21	66.58

Table H.8: **Comparison to CLIP prompting.** The prompt decomposition of VideoGEM is compared to the prompting technique of CLIP. The 80 prompt templates that are provided by the CLIP paper are used to generate 80 prompts. These prompts are used in two different settings. In the first setting, the 80 prompts are averaged, resulting in one text embedding. The combined weights of VideoGEM are then applied (*avg. txt.*). In the second setting, the combined weights are applied to each prompt individually resulting in 80 heatmaps (*avg. heat.*). The final prediction in both settings is obtained by taking the location of the maximum logit in the heatmap. VideoGEM is applied with ViCLIP as a backbone on video input. Both models are evaluated on V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT).

H.2.6 Ablation for CLIP Prompting

We compare the prompt decomposition of VideoGEM with the prompting technique proposed by CLIP in Table H.8. We apply the 80 prompt templates that are provided by the CLIP paper to the test labels. Two different settings are used to obtain the final prediction. In the first setting, the text embeddings are averaged (*avg. txt.*) and then the combined weights of VideoGEM are applied to the single averaged text embedding resulting in a single heatmap. In the second setting, the combined weights are applied to each of the 80 prompts individually resulting in 80 heatmaps that are averaged pointwise (*avg. heat.*). The location of the maximum logit in the heatmap is the predicted location for both settings, similar to VideoGEM. VideoGEM significantly outperforms CLIP's prompting technique. This shows, that the benefit of our proposed prompt decomposition comes not only from the majority voting but the prompt decomposition itself.

H.3 QUALITATIVE ANALYSIS

We present qualitative examples of predictions from our VideoGEM model, using ViCLIP as the backbone with video inputs on the V-HICO and GroundingYouTube datasets in Figures H.2 and H.3 respectively. The ground truth bounding box is green and the final VideoGEM prediction is white. The individual prompt predictions and heatmaps are shown for action, object, and verb prompts in red, blue, and yellow respectively. We observe that the heatmaps

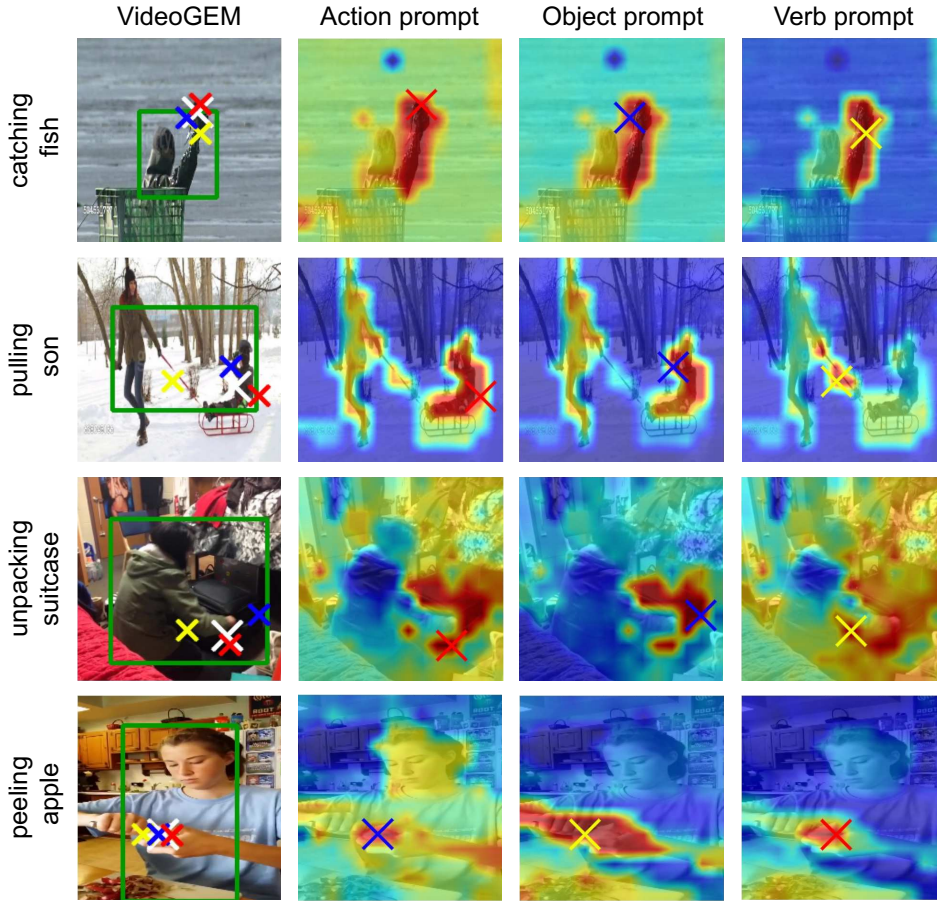


Figure H.2: **Qualitative examples for VideoGEM on V-HICO.** VideoGEM is applied with ViCLIP on video data. The main frame with its label is shown. The green bounding box is the ground truth and the white cross is the final prediction of VideoGEM. Besides that are the heatmaps for the action, object, and verb prompt. The individual predictions for the action, object, and verb prompt are shown by red, blue, and yellow crosses respectfully. The ground truth label of the image is shown on the left.

and the predicted locations for verb prompts differ the most from action and object prompts. Mostly all three predicted locations are close together, especially for actions with small spatial scale such as “*spread butter*”. However, when the action is bigger as for “*unpacking suitcases*” VideoGEM centers the action between its components. Moreover, if predictions are slightly off, such as in “*catching fish*”, or “*pulling son*”, where the action prompt initially is outside the ground truth bounding box focusing only on the object, VideoGEM drags the prediction back into the bounding box, leading to more robust and accurate predictions.

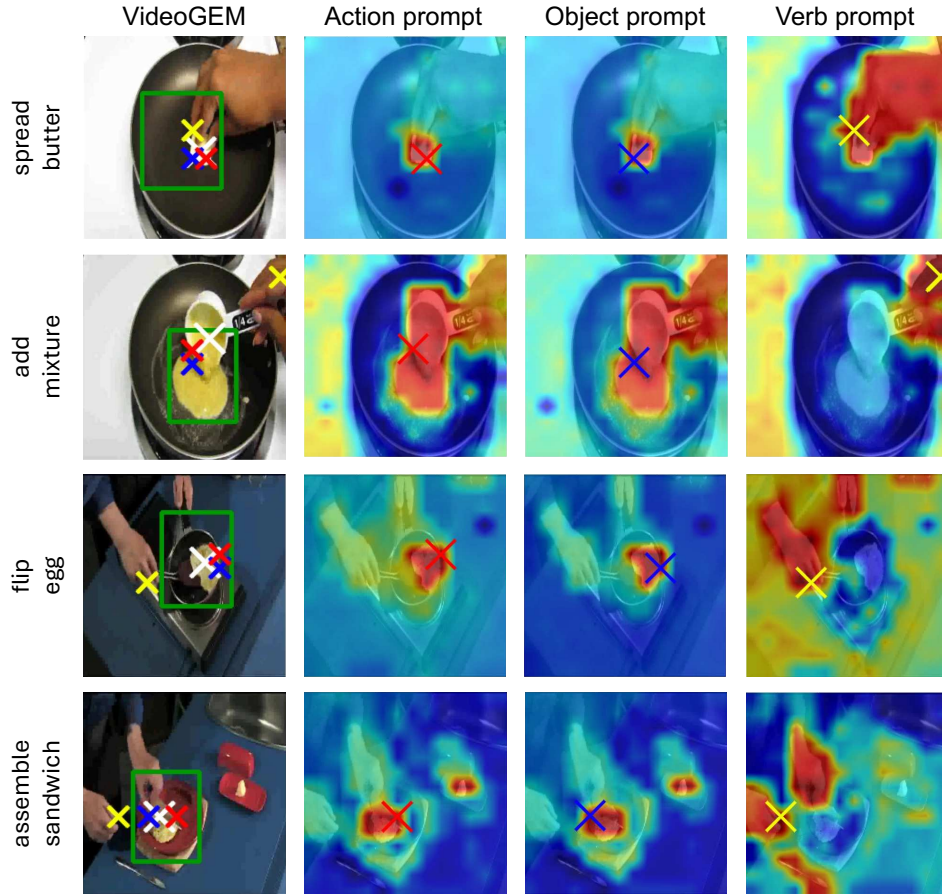


Figure H.3: **Qualitative examples for VideoGEM on GroundingYouTube.** VideoGEM is applied with ViCLIP on video data. The main frame with its label is shown. The green bounding box is the ground truth and the white cross is the final prediction of VideoGEM. Besides that are the heatmaps for the action, object, and verb prompt. The individual predictions for the action, object, and verb prompt are shown by red, blue, and yellow crosses respectively. The ground truth label of the image is shown on the left.

H.4 LIMITATIONS

VideoGEM is designed for spatial action grounding. It considers the temporal context of videos only implicitly via its video backbone. VideoGEM’s predictions are spatial locations for the given input images/videos without temporal predictions. Moreover, the proposed prompt decomposition only works for action grounding in its current state. However, this could be adapted to more general settings.

