

Robust Detection of Out-of-Distribution Data

Dissertation

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

vorgelegt von
M.Sc. Julian Bitterwolf
aus Schwäbisch Hall

Tübingen
2024

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

Tag der mündlichen Qualifikation: 18.09.2024

Dekan: Prof. Dr. Thilo Stehle

1. Berichterstatter: Prof. Dr. Matthias Hein

2. Berichterstatter: Prof. Dr. Philipp Hennig

This work is licensed under a CC BY 4.0 license © ⓘ.

To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0>.

Abstract

Deep neural networks, trained on large amounts of data, have become a highly successful tool for a variety of cognitive tasks. In many of those, they exceed human performance in several aspects, particularly when speed and economic costs are considered. Extrapolating from the quickly ongoing progress in the field of machine learning, which is driven both by advances of algorithmic techniques and by continuously growing scales of available computing hardware, it is anticipated that the amount of analytical and creative tasks that can be solved effectively and efficiently by deep learning models will keep increasing rapidly. This swift progress is already going hand-in-hand with the deployment of deep learning based models in more and more practical applications. Many of these involve usage in situations that are safety-critical or of otherwise great importance for affected individuals and groups, whether they actively use the model or are merely passively affected by its outputs. This makes robustness against input-based issues that can come up once the model is deployed a quintessential aspect of deep learning models and a prerequisite to considering them as adequate tools for such tasks.

In this dissertation, we explore and improve the behaviour of deep neural network image classifiers when confronted with out-of-distribution (OOD) samples, which are inputs that are unrelated to anything the model has been shown during training. A typical failure mode is that classifier models tend to very confidently predict some of those OOD samples to belong to one of the available classes, even though that class is not present in the image sample.

We start by analyzing methods that robustify models against the occurrence of OOD samples by helping to detect them. To do this, we first discuss which test inputs are and aren't OOD in a sense that is compatible with the common evaluation metrics for OOD detectors. Motivated by finding that for the popular ImageNet classification dataset, many existing dataset options for OOD detection evaluation fail these criteria, we introduce with NINCO a new dataset of carefully selected OOD images, which allows us to precisely evaluate a multitude of OOD detectors. We then uncover theoretical similarities and differences between OOD detection methods, and confirm these relationships experimentally. In this context, we find a remarkably apparent dependence of the performance on OOD detection and classification tasks on the sharing of model weights and therefore learned representations between those tasks.

After the evaluation and theoretical analysis of OOD detection methods, we observe that they quickly become helpless when very slight, but worst-case noise perturbations are added to the out-of-distribution input images by an adversarial attack. To alleviate this worst-case risk, we first train models that are empirically robust to such adversarially perturbed out-of-distribution inputs, by devising the ACET method. Subsequently, we construct the GOOD and ProoD models which are even provably invulnerable to these perturbations, remarkably without impeding the models' performance on their main image classification task.

Kurzfassung

Auf großen Datenmengen trainierte tiefe neuronale Netze (deep neural networks) sind in jüngster Vergangenheit zu einem überaus erfolgreichen Instrument zur Bearbeitung einer Vielzahl kognitiver Aufgaben herangewachsen. In vielen Bereichen übertreffen sie von Menschen erbrachte Leistungswerte, insbesondere wenn Bearbeitungsgeschwindigkeit und wirtschaftliche Kosten in die Betrachtung miteinbezogen werden. Wenn man den derzeitigen raschen Fortschritt im Entwicklungsfeld des maschinellen Lernens extrapoliert, dessen Triebkraft einerseits Forschungsfortschritte in der Konstruktion einschlägiger algorithmischer Methoden und andererseits die sich kontinuierlich steigernden Größenordnungen verfügbarer Rechenkapazitäten sind, so ist ein massiv wachsender Umfang der verarbeitenden sowie kreativen Aufgaben zu erwarten, die von solchen neuronalen Netzen zukünftig effektiv und effizient übernommen werden können. Diese zügige Weiterentwicklung des Machbaren geht bereits jetzt mit dem Einsatz lernbasierter Systeme in einer steigenden Zahl praktischer Einsatzbereiche einher. Viele solcher Anwendungen schließen Situationen ein, welche sicherheitsrelevant oder auf sonstige Weise von großer Bedeutung nicht nur für solche Einzelpersonen oder Gruppen sind, die das System aktiv einsetzen, sondern auch für jene, die lediglich passiv von seinen Ergebnissen betroffen sind. Ihre zunehmende Bedeutung macht die Robustheit auf neuronalen Netzen basierender Systeme gegenüber einer alarmierenden Vielzahl von Problemstellungen, die sich aus Unregelmäßigkeiten der ins Modell eingespeisten Daten ergeben können, sobald es als Teil eines fertigen Systems eingesetzt wird, zu einem essenziellen Aspekt des Modells. Derartige Robustheit zuverlässig aufzuzeigen ist in einer beträchtlichen Reihe von Aufgabenbereichen eine notwendige Voraussetzung, damit ein lernbasiertes Modell als adäquates Hilfsmittel in Betracht kommen kann.

In dieser Dissertation inspizieren und verbessern wir das Verhalten neuronaler Netze, welche zur Klassifizierung von Bildern in verschiedene Kategorien entworfen und trainiert wurden, im Falle ihrer Konfrontation mit OOD-Bildern. Als OOD (out-of-distribution) bezeichnen wir hierbei Bilder, deren Inhalt nicht zu einer der vordefinierten Kategorien der Klassifizierungsaufgabe gehört und die sich somit auch wesentlich von den Daten, auf denen das Netz trainiert wurde, unterscheiden. Ein typisches Fehlverhalten auf solchen OOD-Bildern besteht darin, dass Klassifizierungsmodelle für diese Bilder häufig sehr hohe Wahrscheinlichkeitsschätzungen abgeben, dass in ihnen ein Objekt einer bestimmten Kategorie zu sehen ist, was natürlich in Wirklichkeit bei OOD-Bildern nach Definition nicht der Fall ist.

Zum Einstieg analysieren wir Methoden, welche dazu gedacht sind, die Robustheit gegenüber dem Vorkommen von OOD-Daten zu festigen, indem diese als solche detektiert werden. Hierzu führen wir eine klare Definition ein, die festlegt, welche Bilder eindeutig OOD bzw. nicht OOD sind, und welche es ermöglicht, gängige Evaluationsmaße für OOD-Detektoren

zuverlässig auszuwerten und aus ihnen aussagekräftige Schlüsse zu ziehen. Wir beobachten, dass für ImageNet als Grundklassifizierungsaufgabe viele der existierenden Datensätze, die in der Literatur zur Auswertung des OOD-Verhaltens herangezogen werden, die Kriterien dieser Definition verletzen und somit nicht als hinreichend OOD angesehen werden sollten. Um diese Situation zu verbessern, stellen wir mit NINCO einen von uns selbst kuriierten OOD-Datensatz aus individuell sorgfältig ausgewählten OOD-Bildern vor. Dieser Datensatz erlaubt uns die präzise Auswertung einer Vielzahl gängiger OOD-Detektoren. Darauf folgend erschließen wir eine Reihe theoretischer Verbindungen und Abgrenzungsmerkmale unterschiedlicher OOD-Detektionsmethoden, welche wir in Experimenten auf verschiedenen Datensätzen bestätigen. In diesem Zusammenhang registrieren wir eine erstaunlich deutliche Abhängigkeit der Klassifizierungs- und OOD-Detektionsleistung eines Modells davon, ob seine gelernten Parameter und damit seine internen Datenrepräsentationen für beide Zwecke einzeln oder gemeinsam trainiert wurden.

Die bis dahin betrachteten Methoden zur OOD-Detektion brechen, wie wir zeigen, sehr leicht zusammen, wenn kaum sichtbare, aber für die Umgehung des Detektors optimierte, geringfügige Manipulationen an einem OOD-Bild vorgenommen werden. Um Risiken durch derart stark begrenzte, aber schlimmstmögliche Bildveränderungen zu beschränken, trainieren wir zuerst unsere ACET-Modelle, welche OOD-Daten, nach empirischen Angriffsversuchen zu urteilen, robust detektieren können. Anschließend konstruieren wir mit GOOD und ProoD Modelle, die beweisbar gegen Auswirkungen derartiger Bildmanipulationen geschützt sind. Bemerkenswerterweise erhalten wir diese beweisbaren Robustheitszertifikate, ohne dabei negative Auswirkungen auf die Klassifizierungsleistung der Modelle auf unveränderten Bilddaten in Kauf nehmen zu müssen.

Acknowledgements

I consider myself incredibly fortunate to have shared the intense and formative journey of working towards this dissertation surrounded by so many wonderful people. Not only were all of them excellent colleagues and collaborators; many became close friends, and I cherish the bonds we've built that will definitely endure long beyond this academic milestone.

I am particularly grateful to my excellent advisor, Matthias Hein, for his invaluable guidance. His profound understanding of the field and his great ability for pinpointing crucial aspects of a project's current state, whether intriguing or requiring clarification, continuously proved instrumental towards producing the works included in this dissertation. He instilled in me the importance of focusing on relevant and impactful research questions, and I learned an immense amount from him.

I'm immensely grateful to Alex Meinke, not only for being an excellent collaboration partner making our work on several conference papers both enjoyable and fruitful, but also for being an engaging discussion partner on a magnitude of topics, some of which turned into successful new approaches and improvements, and all of which for me resulted in a deeper understanding of topics ranging from probabilistic aspects in certain machine learning algorithms to the very nature of reality. I am also very thankful to Václav Voráček for more than adequately filling the deep-discussion-void, which was left behind by Alex at some point, with frequent scientific arguments, always bringing forward thoughtful, deeply informed, and refreshingly strong opinions which usually turned out to be surprisigly correct, and most importantly for consistently being an enthusiastic reminder about most important ingredient to successful research: having fun with it, especially when uncovering and tackling mathematical problems.

Another special thank you goes to Max Augustin, a great source of wisdom on all practicalities of state-of-the-art vision models and on how to actually have them run on hardware, as well as on the hot developments and emerging limitations in the broad and ever-evolving world of machine learning. I'm also grateful for the joy of collaborating with Maxi Müller, working with whom made even the experience of raking through thousands of images in order to find certain incidental objects in the background an enjoyable activity. I also want to express many warm thanks to Pedro Mercado Lopez for helping me cope with the scary aspects of being a fresh doctoral student and resident of Tübingen, for sharing many prudent approaches to advancing in typical situations that occur during research projects, and for always being such a nice and helpful person be around.

I would like to address many thanks to Valentyn Boreiko, for being a fantastic and always encouraging colleague and for all the stimulating conversations regarding the ups and downs of all aspects of research, career choices and life. I further want to thank Maksym

ACKNOWLEDGEMENTS

Andriushchenko, who as a collaborator on my first PhD project with his sensational technical and scientific proficiency helped me learn a lot about the important methods in robust deep learning. Many thanks also go to Francesco Croce for being an excellent colleague and a walking well of robustness knowledge. I also sincerely thank Roland Zimmermann, for being a great workshop co-organizer and for many thought-provoking debates. I further want to thank Patrick Ferber, Jörg Hoffmann and Daniel Heller for being excellent collaboration partners and guiding me through the gridworlds of AI planning.

I am extremely grateful for the great social environment brought about by all members of our research group, and I want to expressively thank Naman Singh, Christian Schlarmann, Yannic Neuhaus and Amit Peleg for all the consistently inspiring discussions during work days and for making the many recreational meetings as delightful and entertaining as they were. This gratitude extends to the many excellent colleagues in the whole Tübingen AI Center. I also want to extend many thanks to Barbara Schiffner, Charlotte Wenner and Jessica Endress for always being very helpful and taking so much bureaucratic work off everyone's shoulders. I further thank Philipp Hennig, Matthias Hein, Seong Joon Oh and Wieland Brendel for taking the time to examine this dissertation and its defense. Additionally, I want to extend my sincere appreciation to those positive aspects of the academic system which provides a wonderful platform for learning researchers like myself to pursue questions that are fascinating to them. Particularly the University of Tübingen which I also acknowledge together with the BMBF, the DFG and the Tübingen AI Center for providing funding for my PhD research position. Arriving at this point would also not have been possible without the fantastic academic training I received from KIT and École Polytechnique during my previous studies. Of course, I am keenly aware and deeply grateful for the contributions of the taxpayers, which make accessing these academic institutions and enjoying their scientific freedoms possible.

I can only try to express my most immeasurable gratitude towards my loving partner Evgenia Rusak. I can hardly imagine completing this doctorate without her being the fixed star by my side, not only making every day brighter, but also by being there as continuous support in emotional, academic and all other matters. Finally, I am also immensely thankful to my parents, brothers, grandparents and extended family as well as all my friends for being constant source of strength throughout my academic journey.

Contents

Abstract	iii
Kurzfassung	v
Acknowledgements	vii
1 Introduction	1
1.1 Input data are not always identically distributed	1
1.2 Out-of-distribution detection	3
1.2.1 OOD detection metrics	5
1.3 Adversarial robustness	6
1.3.1 Adversarial attacks on image classifiers	7
1.3.2 Adversarial attacks on OOD detectors	8
1.3.3 Empirical adversarial robustness	9
1.3.4 Certified adversarial robustness	10
1.4 Outline	12
I The Out-of-Distribution Detection Problem	15
2 When is an Image Out-of-Distribution?	17
2.1 Requirements for an OOD dataset	17
2.2 Issues with previously existing OOD and OSR datasets for ImageNet-1K . .	21
2.2.1 Prevalence of ID samples in popular OOD datasets	22
2.2.2 Effect of ID contamination on OOD evaluation	22
2.3 NINCO : A new OOD test set for ImageNet-1K	24
2.3.1 NINCO dataset construction	26
2.3.2 OOD unit-tests	27
2.4 OOD detectors for ImageNet and how to evaluate them	28
2.4.1 OOD detection methods	28
2.4.2 Evaluating OOD detectors on NINCO	31
2.5 Evaluation results for IN-1K OOD Detectors	32
2.5.1 Results on NINCO	33
2.5.2 Results on the OOD unit-tests	35
2.6 Detailed results on NINCO	36

2.7	Chapter conclusions	37
II	The Building Blocks of Out-of-Distribution Detectors	41
3	Breaking Down Out-of-Distribution Detection	43
3.1	Goals for the analysis of OOD detectors	44
3.2	Models for OOD data and equivalence of OOD detection scores	45
3.3	Bayes-optimal behaviour of binary discriminators and common OOD detection methods	47
3.3.1	OOD detection with methods using unlabeled data	47
3.3.2	OOD detection for methods using labeled data	49
3.3.3	Separate vs. shared estimation of $p(i x)$ and $p(y x, i)$	56
3.4	Confidence loss models and models with a background class or a binary discriminator are only equivalent after s_3 is applied	58
3.4.1	Example problem: Euro coin classifier	58
3.5	An overview of CIFAR scale OOD detection methods	59
3.6	Experiments	61
3.6.1	Out-distribution aware training with labeled in-distribution data	63
3.6.2	Shared representation learning for the binary discriminator	64
3.7	Chapter conclusions	64
III	Adversarially Robust Out-of-Distribution Detection	69
4	Empirically Adversarially Robust Out-of-Distribution Detection	71
4.1	Goals and overview	71
4.2	ReLU networks produce piecewise affine functions	72
4.3	Why ReLU networks produce high confidence predictions far away from the training data	74
4.4	ACET: Adversarial Confidence Enhanced Training	77
4.5	Experiments	79
4.5.1	Training	82
4.5.2	Evaluation	83
4.5.3	Main results	84
4.5.4	Far away high confidence predictions	85
4.5.5	Image level effects of ACET on ID and OOD confidences	85
4.6	Chapter conclusions	86
5	Certifiably Adversarially Robust Out-of-Distribution Detection	89
5.1	Motivation for worst-case OOD detection guarantees	89

5.2	Definitions, setup and baselines	90
5.2.1	AUC and Conservative AUC	92
5.3	GOOD: Provable guarantees for OOD detection	92
5.4	Experiments	95
5.4.1	Model architectures and data augmentation	96
5.4.2	GOOD training procedure	96
5.4.3	Evaluation setup	98
5.4.4	Experimental results	99
5.5	Chapter conclusions	101
6	Separation of Certified and Standard OOD detection	107
6.1	Improvement goals for certified worst case OOD detection	107
6.2	ProoD: The Provable out-of-Distribution detector	110
6.2.1	Joint Model for OOD Detection and Classification	111
6.3	Guarantees on asymptotic confidence	114
6.4	Experiments	116
6.4.1	Training of ProoD	116
6.4.2	Evaluation	117
6.5	Adversarial asymptotic overconfidence	123
6.6	Chapter conclusions	124
7	Conclusions and Outlook	125
	Abbreviations	130
	Bibliography	131

Chapter 1

Introduction

Models based on deep learning exhibit impressive performances on a growing range of cognitive tasks and play an increasingly important role in many aspects of society, industry, and people’s daily lives. These models are typically obtained by fitting the parameters of a deep neural network (DNN) on large amounts of training data. The tasks where deep learning based methods most excel compared to other approaches like rule based algorithms usually entail highly complex data like photos (Krizhevsky et al., 2012), texts (Collobert et al., 2011), genetic information (Xiong et al., 2015; Libbrecht & Noble, 2015), 3D MRI scans (Lundervold & Lundervold, 2019) and many others for which sensible patterns and informative features cannot be found by manual engineering combined or applying basic statistics.

The stochastic nature of the training process, notably including the randomness of the collection of training data, combined with the complexity of the resulting trained deep neural network, often prohibits making reliable statements about the model’s behaviour on unseen inputs. One commonly considered issue—of which users of systems that involve deep learning are often quite aware—lies in the near impossibility to predict with certainty what a model will output when given a specific input sample, until we actually evaluate the model on the sample. Reducing the probability of making errors on such data points that are unseen, but still similar to what a model has been trained on, is a standard objective in machine learning and often regarded as a central measure for progress in this field. However, when applying them in the wild, additional issues tend to arise since the models frequently find themselves in an open-world setting that considerably differs from their training regime: Besides the consideration of generalization to *individual samples*, a conceptually similar problem occurs when applying the model on *new types of inputs* that are of dissimilar nature compared to those it has been trained on or previously tested and found suitable for. In this dissertation, we focus on the latter issue, which we expand on in the following paragraphs, and we will use image classification models to illustrate and experimentally analyse the treated topics.

1.1 Input data are not always identically distributed

A commonly made, standard assumption when working with machine learning methods is that training and test data be *independent and identically distributed* (i.i.d.), which means that the training samples are representative of the inputs that will be encountered when the trained

classifier is deployed for its intended use. Under this assumption, a primary interest is to train a model on the limited number of examples in the training dataset in such a way that it generalizes well to individually unseen test data points. Hence test accuracy, which is the typical measure of this in-distribution generalization, is one of the main metrics that is commonly used to assess the progress of deep learning methods on image classification tasks. However, a critical issue when applying such trained models is that they often exhibit unforeseen behaviour when confronted with inputs of which the training set is not representative.

There are multiple discernible ways in which such non-identically distributed inputs can occur and pose problems to image classifiers that were designed and trained under the i.i.d. assumption. For one, a classifier model might encounter objects that do not belong to any of the classes that it has been trained for. For example, an animal or foreign street sign that does not appear in the training data might be encountered by an autonomous vehicle, a patient might have a very infrequent disease that does not appear in the training dataset for a medical diagnosis application (Zhou et al., 2021; Roy et al., 2021), or a technical error might result in a camera providing only black images to the model in an industrial process monitoring system. Throughout this dissertation the term “*out-of-distribution*” (OOD) specifically refers to this kind of non-i.i.d. input; we will discuss its definition in detail below. As we will highlight on several occasions, models built under the i.i.d. assumption in many cases behave erratically on such OOD inputs.

In another form of distribution shift, the images encountered at test time still show objects that belong to one of the available classes, but some characteristics of these images differ from what the model has been trained on. For example, the training set might consist of well lighted images taken during sunny daytime, and later the model is applied in inconsistent conditions like in snowy weather or at night (Hendrycks & Dietterich, 2019), image characteristics might differ starkly between different regions on earth (Koh et al., 2021), or a medical imaging setup might be calibrated differently in different hospitals (Maier-Hein et al., 2022). Other types of distributional shift include different class prevalences between training and test data and, in a wider sense, errors or inconsistencies of training labels compared to the desired test time predictions.

Besides naturally occurring non-i.i.d. test samples, a notorious robustness issue for deep neural networks is the existence of adversarial examples (Biggio et al., 2013; Szegedy et al., 2014). These demonstrate that even tiny amounts of added noise can change the decision of a classifier for an input when the noise perturbation is deliberately chosen to provoke this changed result. It is possible to train models that exhibit robustness against adversarial perturbations as long as those are small enough. This, however, usually comes with a trade-off in performance on benign, i.i.d. inputs, and the robust radii that have been achieved (Croce et al., 2021) for standard scale image classification like ImageNet-1K (Russakovsky et al., 2015) are still too small to cover most humanly virtually imperceptible adversarial perturbations. Whether robustness to adversarial examples, as well as to OOD inputs or distributionally shifted samples, is solvable is not immediately clear from a technical perspective. However, humans usually have no hard time correctly classifying or detecting such inputs. While it can be argued that humans do not have to work with limited training data, and might be vulnerable

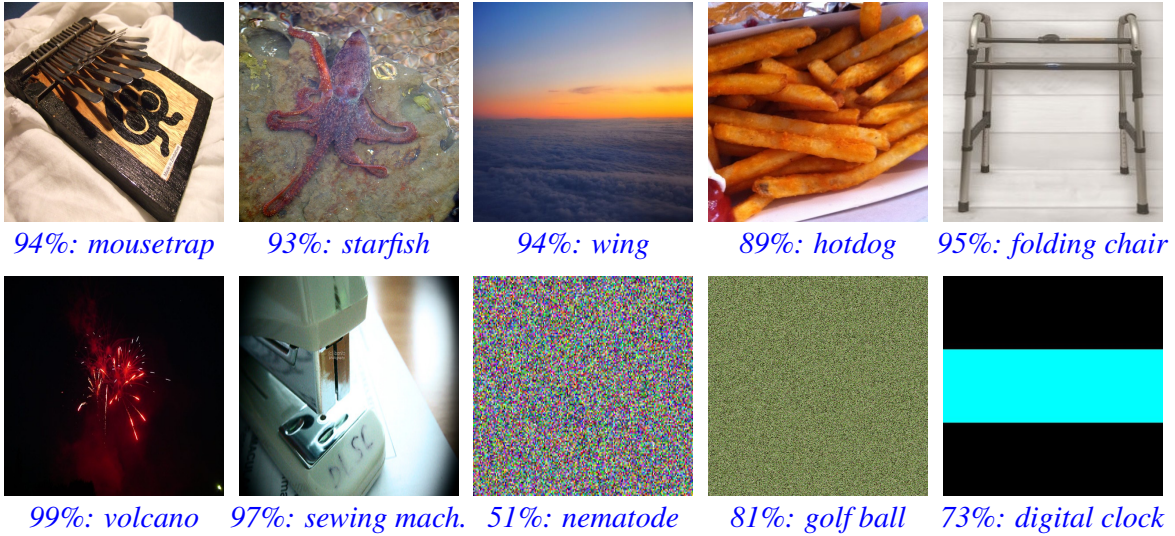


Figure 1.1: **High confidence predictions on OOD samples made by a deep neural network classifier for the ImageNet-1K dataset.** The classifier is a ViT-B-224 with a standard accuracy of 84.5% and a mean confidence of 84.7% on the ImageNet-1K validation set. The images show: *mbira* (musical instrument), *ruby octopus*, *sky*, *fries*, *walker frame*, *fireworks*, *stapler*, *uniform noise*, *permuted pixel noise*, *tricolour stripes*, none of which fit into any of the 1000 ImageNet-1K classes.

to other inputs that are not found as easily as for deep neural networks, they still demonstrate that the respective specific robustness issues can in principle be overcome. Furthermore, the seemingly effortless human capabilities with respect to those robustness issues may make these shortcomings of models that might exhibit better than human performance on some i.i.d. tasks particularly surprising and thus important to soundly investigate before assuming that a model will do well in the real world.

1.2 Out-of-distribution detection

Deep learning models have been observed to display erratic behaviour when confronted with unknown situations like receiving an input that is not related to the task it has been trained on. Such samples are what we call *out-of-distribution* (OOD). Deep neural network classifiers are known to frequently make very confident predictions, claiming that those samples belong to one of the *in-distribution* (ID) classes (Hendrycks & Gimpel, 2017; Hein et al., 2019; Nguyen et al., 2015). We illustrate a standard, high performance image classifier’s high confidences, by which we generally mean the maximum predicted probability in any of the classes, on OOD inputs in Figure 1.1. For example, the classifier expresses a 94% certainty in its assessment a mousetrap is shown on the image of an *mbira*, which is a traditional Zimbabwean instrument that does not appear in the 1000 classes of ImageNet-1K (Russakovsky et al., 2015), the

dataset on which the model has been trained. This unwanted behaviour is a serious obstacle when applying classifiers in the real world, since in many applications it is inevitable that at some point a semantically unexpected type of input is encountered, especially if the model is employed outside of its exact training environment.

To address this problem, one can deploy OOD detection methods, whose purpose is to reject such OOD inputs while letting only ID inputs pass. Depending on the application, such a reject decision can mean requesting human intervention, steering towards a safe state, or simply abstaining from making a prediction, while at the same time letting ID inputs pass through.

The task of OOD detection is closely related to those of *anomaly detection* (AD) and *open set recognition* (OSR), with the boundaries often being blurred in the literature. AD, which has long been an intensively studied research topic in statistics (Chandola et al., 2009), in the image recognition context usually refers to the setting where we do not have an explicit, labelled class structure on the in-distribution. OSR (Bendale & Boulton, 2016b), on the other hand, is explicitly interested in simultaneously detecting OOD inputs as well as accurately classifying ID samples, which makes the setting very similar compared to OOD detection.¹

Within OOD detection, we will regard two distinct scenarios: in the basic OOD detection setting, only standard ID training data is available, and in many cases detection methods can be applied *post-hoc*, meaning that they are applied to a standard trained “off the shelf” classifier. The other setting assumes the availability of *surrogate OOD training data*, which is not assumed to be related to the unseen OOD inputs that the classifier may encounter later, but that are used to give the model “a sense” that there is something outside of the ID to be cautious about.

A well studied research area that explicitly aims at avoiding overconfidence on standard ID data is classifier calibration (Guo et al., 2017), which aims for the output class predictions to accurately reflect the distribution wide probability of the input belonging to the respective class. Since we introduced the classifier overconfidence on OOD data as the motivation problem for OOD detection, it would be natural to try to resort to existing calibration methods, among others using softmax calibration via post-procession rescaling (Guo et al., 2017), ensemble techniques (Lakshminarayanan et al., 2017) or Bayesian approximation techniques, e.g. using dropout (Gal & Ghahramani, 2016). However, these methods mostly have no significant positive impact for reducing the relative confidence on OOD inputs compared to ID data (Leibig et al., 2017). Rather, in recent years, a range of methods specifically for improving OOD detection have been proposed. For example, Mahalanobis OOD detectors (Lee et al., 2018b) fit a distance on the neural activations of a standard trained classifier and mark data as OOD if they are far away from ID classes when measured in this distance. Energy based OOD detection (Liu et al., 2020) and Max-Logit (Hendrycks et al., 2022) use an observation that logit values of ID data often tend to be higher than those of OOD inputs. Outlier Exposure (Hendrycks et al., 2019a) and CEDA (see Chapter 4) train a model to have low confidence

¹Note that in the literature occasionally different definitions and standard settings are used, e.g. (Scheirer et al., 2012) additionally presume the availability of positive and negative training samples for OSR.

on surrogate OOD data, with the former using natural images and the latter using synthetic noise as surrogate outliers. A wide number of further OOD detection methods, several of which are quite similar to the ones mentioned here, have been proposed in the literature. We provide summaries of various of these methods in the chapters where we also evaluate them, with extensive lists on post-hoc methods in Chapter 2, and on methods using surrogate OOD training data in Chapter 3.

Besides OOD detectors based on classifiers, it appears natural to use a generative or density based model like a VAE or GAN to model the in-distribution and distinguish its test samples from OOD inputs. While these approaches showed promising results for some OOD datasets, using them for general OOD detection has shown to be challenging. Several works (Nalisnick et al., 2019; Hendrycks et al., 2019a) have reported that generative approaches can produce highly confident density estimates also for inputs outside of the class they are supposed to model. In Chapter 3, we formalize density based OOD detection approaches and provide a possible explanation for their observed shortcomings.

1.2.1 OOD detection metrics

The practical purpose of an OOD detector is to make the binary decision whether an input is deemed ID and thus accepted, or OOD and thus rejected. In this sense, it is similar to a classifier which predicts the input to belong to a specific class. Rather than having a model make the full accept/reject decision, one commonly regards OOD detectors that output an *OOD detection score* for a given input, which is generally supposed to be larger on ID inputs than on OOD inputs. While classifiers often produce an output vector that can be interpreted as the predicted class probabilities, the fraction of OOD samples per ID sample to be expected when applying the detector is generally not given. Hence, this score can not readily be converted into a probability. Further, the occurrence of OOD inputs at test time may be rare, but the cost of missing an OOD sample or wrongly rejecting an ID one is very dependent on the application. Because of these circumstances, in OOD detection we cannot simply measure the detector’s accuracy and will use other metrics to assess the usefulness of its score outputs. In practice, the decision will usually be based on a fixed threshold τ , such that the detector rejects any sample for which it calculates a score below τ and accepts those which reach a score of at least τ . The fraction of samples from an ID test set that receives a score $\geq \tau$ measures the OOD detector’s *true positive rate* (TPR), and the fraction of mistakenly accepted OOD inputs is the *false positive rate* (FPR) for the respective OOD test set.

One common metric for evaluating OOD detection performance is the *false positive rate at a fixed true positive rate* Q , denoted as $FPR@TPRQ$. Here, τ is fixed on the ID test set such that the TPR equals Q , and for a given OOD test set, one reports the FPR at this τ . Since the TPR in expectation holds for any set of new ID samples, $FPR@TPRQ$ represents the failure probability of a fixed threshold detector which is calibrated to achieve the desired TPR of Q . Thus, lower $FPR@TPRQ$ indicates better OOD detection performance. In the OOD detection literature, the most commonly used value for Q is 95%, which we too use throughout this dissertation.

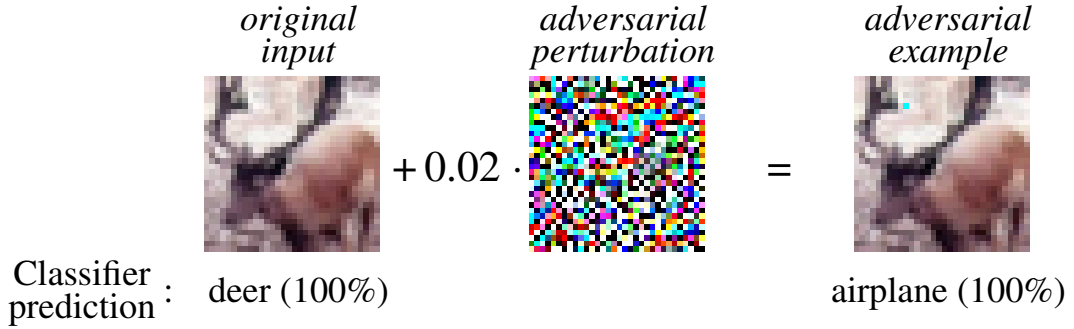


Figure 1.2: **Adversarial attack on the classification decision:** The standard classifier for the CIFAR-10 dataset correctly identifies the original image to the left as a deer, with 100% confidence. In the center image, a small adversarial perturbation of size $\epsilon = 0.01$ (the maximum pixel color value increase or decrease) is visualized by increasing its color intensity by a factor of 50 and adding it to a gray image. Adding this perturbation to the original image changes the classifier’s decision into the incorrect class ‘airplane’.



Figure 1.3: **Confidence increasing adversarial attack:** A classifier trained with Outlier Exposure (Hendrycks et al., 2019a) prudently assigns low confidence to the image of a sunflower, which is not a class of CIFAR-10: the predicted probabilities of each of the 10 classes is at most 10.3%. However, a slight perturbation is sufficient to fool the model into predicting ‘cat’ with very high confidence.

Another useful metric is the *area under the receiver-operator characteristic curve*, short *AUROC* or just *AUC*. It represents the probability that an ID input receives a higher score (equal scores counted half) than an OOD input when both are drawn randomly from their respective evaluation datasets. Alternatively, it can be regarded as the mean FPR@TPRQ when Q is varied over all possible values between zero and one.

1.3 Adversarial robustness

While modern deep neural network image classifiers achieve remarkably low test error for many tasks, it is in many cases easily possible to deliberately produce input images that are misclassified (Biggio et al., 2013; Szegedy et al., 2014). These *adversarial examples* are

obtained by subtly perturbing an image such that the classifiers decision changes while a human beholder does not perceive that the image has been manipulated. While there are many different threat models that approximate the set of allowed, imperceptible perturbations, in this work we are focusing on so-called l_∞ perturbations: Each RGB color value of each pixel can be increased or decreased by up to a value of a certain ϵ , which typically is a small fraction of the total color intensity range. As we see in Figure 1.2, for a human observer the semantic content of the accordingly perturbed example image does not change, while the DNN classifier completely and confidently changes its prediction. While it has been observed that adversarial examples can contain some small scale, but in principle informative visual features of the incorrectly predicted class (Ilyas et al., 2019), and that adversarial perturbation can even affect human decisions in certain scenarios (Elsayed et al., 2018; Veerabadran et al., 2023), in most cases one can argue that humans still make the correct decision on those images, which attests that robustness can in principle be achieved. Thus, it is mostly agreed that the existence of adversarial examples shows an existing weakness of deep neural networks, and that robustness against those minuscule adversarial perturbations is a desired property for a model, particularly if it does not come at a cost e.g. lower accuracy on unperturbed inputs.

1.3.1 Adversarial attacks on image classifiers

An *adversarial attack* is a process that, starting from a given original input sample, aims to find an adversarial example which is permitted by the considered threat model. In our case this means that we regard l_∞ bounded attackers for a given radius ϵ . The goal of an adversarial attacks on a multi-class classifier can be either *targeted*, attempting to change the decision towards a specific outcome, or *untargeted*, with the sole objective of changing the decision into an incorrect one. For simplicity, we limit the following exposition to the untargeted attack objective. An important distinguishing characteristic between adversarial attacks is what kind of access they have to the models decision process. *White-box attacks* assume full knowledge of the parametrized model, including the ability to perform backward passed in order to calculate loss gradients with respect to the input pixels. *Black-box attacks*, on the other hand, for each input query only receive the model's output in the form of either class probabilities or even merely the singular class decision. Throughout this dissertation we assume that the attacker has white-box access to the model, since utilizing the strongest available attack allows for the best approximation of the true adversarial robustness. However, it can make sense to use black-box techniques besides gradient based methods even where the latter are available in order to avoid being overly reliant on potentially misleading gradient information.

A basic adversarial attack is the Fast Gradient Sign Method (FGSM) (Biggio et al., 2013; Szegedy et al., 2014) which at the original sample calculates the gradient direction that locally moves the model's output fastest away from the correct class decision. Then each pixel is perturbed by the maximal permitted amount of ϵ or $-\epsilon$, such that the sign of the perturbation is aligned with the sign of the gradient. FGSM is computationally cheap and is often able to find adversarial examples for standard, undefended models, making it an early tool for

highlighting the issue of the existence of adversarial examples. Higher attack success rates can be achieved with the Projected Gradient Descent attack (PGD (Liu et al., 2017; Kurakin et al., 2018)), which performs several steps of FGSM by newly evaluating the gradient at each iteration. While the size of each step is smaller than ϵ , it is possible that a step leaves this permissible perturbation range; in this case, the respective pixel perturbations are projected back to plus or minus ϵ via clipping. Additional increase in an attack’s success rate can often be achieved by running it several times with different randomized parameters or starting points (with the threat model and potential projections still being determined by the original sample).

While combinations of strong PGD variants with a high computational budget for restarts often constitute an attack that reliably finds adversarial examples even for robustly trained models (see 1.3.3), certain defenses deliberately or involuntarily introduce an effect of masked or obfuscated gradients (Athalye et al., 2018b), where gradient directions are not informative or misleading as a descent direction towards the attack’s objective. Due to such gradient obfuscations and other effects, it is often important to attempt finding adversarial examples via adaptive attacks (Carlini & Wagner, 2017; Tramer et al., 2020), which are crafted for the specific model under evaluation and carefully take its defense method into account. Though usually weaker than attacks that take all available information into account, black-box attacks (Liu et al., 2017; Papernot et al., 2017; Brendel et al., 2018; Andriushchenko et al., 2020b), which may include transferring adversarial examples from *other* models to which white-box access is possible, can often be a helpful addition to white-box attacks, especially when gradients of the attacked model do not contain enough useful guidance.

1.3.2 Adversarial attacks on OOD detectors

While OOD detection can—once we fix a test out-distribution—in principle be regarded as a classification problem with two classes (and without training data for one class), the two ‘classes’ which are ID and OOD are not symmetrical concepts, which is in contrast to what is usually assumed for standard classification. Rather, one can regard different potential adversarial robustness issues that may arise due to different scenarios that are associated with their individual practical risks. In this work, the threat model of which we investigate several aspects treats ID samples as benign, but assumes that any OOD input could potentially be adversarially perturbed. There are multiple reasons why we find this asymmetric threat model interesting.

For one, it is not clear whether an adversarially perturbed in-distribution input can and should not still be regarded as in-distribution. While possible adversarial changes do not change the semantic meaning of the image, it can be argued (e.g. considering Ilyas et al. (2019)) that some low level features with semantic meaning are added or removed. If perturbed images contain contradictory features that would not appear on natural ones, e.g. cat fur characteristics on a dog, these images are arguably not ID anymore. Also from the perspective of probabilistic image distributions, encountering an adversarial input (Gilmer et al., 2019) is highly unlikely when sampling from the *in-distribution*, making it at least highly atypical, if not clearly OOD.

On the other hand, when we start with an input image that is not ID due to showing no ID objects at all, we can assume that a small perturbation cannot semantically change the image into showing an ID object, at the very least not in a typical occurrence.

In the baseline setting where a standard classifier’s confidence is used for OOD detection, the considered adversarial attacks receive an OOD input image and attempt to find a perturbation that maximizes the classifier’s confidence. We illustrate the issue of adversarial high confidence OOD examples in Figure 1.3. Note that the class whose predicted probability is maximized is arbitrary, and the attack might find an adversarial OOD example on which the classifier confidently predicts a class that is different from its favourite class at the original OOD sample.

1.3.3 Empirical adversarial robustness

Having identified the existence of adversarial examples as an undesired or even precarious phenomenon, we would like to obtain *adversarially robust* classifiers that are not susceptible to such small perturbations. Usually, the metric that adversarially robust classifiers aim at improving is *robust accuracy*, which is defined as the fraction of test samples that are classified correctly even if for each sample the worst case perturbation as permitted by the threat model is applied. For a general ReLU network, finding, if it exists, an adversarial example within the l_∞ ball around a given sample is NP-complete with respect to the network’s size, as Katz et al. (2017) show. This implies that *verifying* the adversarial robustness, i.e. the non-existence of an adversarial example, is co-NP-complete (Aaronson, 2016), which means it is by current understanding likely to be computationally infeasible. Thus, only a lower bound on the adversarial robustness of a model can be calculated, by running a strong attack on each input sample.

The standard method for trying to obtain adversarially robust classifiers is Adversarial Training (AT) (Goodfellow et al., 2015; Madry et al., 2018). Here, the samples of the training set are replaced with their adversarially perturbed version, which is generated for the current state of the model at each training step by running an adversarial attack like PGD. Since performing one step of PGD requires a full forward and backward pass and thus has the same complexity as a standard training step, AT is usually performed with fewer steps and no additional restarts compared to the attacks used for evaluation. Even with these reductions, AT is multiple times as expensive as standard training in terms of compute and training time.

The robustness measured for AT models has proven to be quite resilient even when probed with more novel and compute intensive evaluation attacks. Currently, the models showing the highest empirical robustness (Croce et al., 2021) are usually still trained with some variations of AT, like changing the attack used during training, varying training schedule (Shafahi et al., 2019; Wong et al., 2019; Andriushchenko & Flammarion, 2020), using a different training loss (Zhang et al., 2019; Qin et al., 2019) or varying other parameters. Furthermore, using additional training data, even if it is synthetic or unlabelled, has proven an effective tool in improving empirical adversarial robustness (Gowal et al., 2021; Peng et al., 2023).

While adversarial training successfully improves empirical robust accuracy, it comes

with the drawback of commonly severely reduced standard accuracy on unperturbed images (Tsipras et al., 2019; Raghunathan et al., 2019), and can introduce some delicate optimization and stability issues (Rice et al., 2020).

An important drawback that is inherent to empirical adversarial robustness is that it does not provide guarantees that the measured robust test accuracy of a model approximates the true robust accuracy, since for any individual test sample, we cannot exclude with certainty the existence of a successful adversarial perturbation. As an extreme case of this issue, some originally promising empirical adversarial have been found out to provide only limited robustness, or no robustness at all (Tramer et al., 2020). For example, in certain cases, effects like obfuscated gradients (Athalye et al., 2018b) can prohibit gradient based adversarial attacks, which usually are among the most testing ones, to successfully find existing adversarial examples.

1.3.4 Certified adversarial robustness

While empirical evaluations provide an upper bound on the adversarial robustness of a given classifier, it would often be desirable to know exactly whether a perturbation that changes the model’s decision does or doesn’t exist. For models with acceptable complexity to achieve even mediocre accuracy on datasets like CIFAR-10 or ImageNet, such exact verification is not achievable with current methods and computational resources. One can however aim to provide a guarantee that no adversarial examples exist for certain test samples. This is a lower bound on the robust accuracy, since for samples where no such guarantees are derived, it is generally not certain whether the opposite holds, i.e. whether there actually exists an adversarial example. As discussed before, it is co-NP-hard and thus generally not feasible to *prove* that there is no adversarial perturbation that changes a classifier’s decision for a given sample. While in principle exact verification approaches like mixed integer programming (Tjeng & Tedrake, 2017) exist, those are only feasible for very small networks and do not achieve meaningful verified accuracies on most commonly regarded datasets. However, it is possible to construct classifiers for which calculating robustness certifications is greatly facilitated. This can be done by choosing an adequate network depth and architecture and applying regularization during model training that enforces suitably sized linear regions (Hein & Andriushchenko, 2017), regularized linear relaxations of propagated polytopes (Wong & Kolter, 2018), Lipschitzness (Anil et al., 2019) or other properties that can later be used to certify lower bounds for the model’s adversarial robustness.

One certification method for l_∞ adversarial robustness that we will make use of in this dissertation is *Interval Bound Propagation* (Gowal et al., 2018), short *IBP*. It assumes that each network layer can be written as multiplying an input vector with a weight matrix, as is the case for example with linear and convolutional layers as well as for common normalization and pooling layers, and also allows for standard activation functions like ReLU. IBP uses the information that any pixel of the an adversarially perturbed input image is constrained to the interval of original plus/minus ϵ and calculates a transformed interval around each of the first layer’s output activations. This transformed interval again represents an upper and lower bound on this activation’s value. The transformation itself is based on the insight that in

multiplying the bounded input with a matrix, a new upper bound the output can be derived by applying the positive elements of the matrix row to only the input's upper bounds, and the negative entries to only the lower bounds. The new lower bounds are obtained via the reversed combinations. This way, the bounds are propagated forward layer for layer, until we obtain lower and upper bounds on each logit's value, and therefore a bound on the worst-case output prediction. We give a detailed mathematical description of IBP in Chapter 5, where we modify the output bound calculation in order to upper bound confidence on OOD inputs.

For standard trained networks, and even for empirically robust ones trained with e.g. AT, the bounds obtained from IBP tend to grow exponentially with each layer, which means that the obtained guarantees on adversarial robustness are practically vacuous. A major reason for this is that IBP provides a straightforward way to calculate lower bounds on robust accuracy, these bounds are generally very loose, partly they do not account for cancelling interactions when considering the worst-case input pixel perturbations after more than one layer. Rather, for receiving meaningful bounds, it is necessary to train the model in a way that is aligned with the eventual application of IBP as the certification method. The relative computational simplicity of IBP in principle allows for a direct minimization of the worst case loss given the IBP output bounds. However, actually training models to minimize the IBP resulting from the threat model's ϵ value proves delicate in practice: Activation bounds for a model at initialization can become too large numerically, extremely large gradient updates can make the training erratic, and on the other hand local minima at very conservative globally constant prediction functions can be reached. Because of these issues, a careful training schedule slowly increasing ϵ from zero and combining the bound based loss with standard, non-adversarial cross-entropy is necessary (Gowal et al., 2018), and further modifications like additionally optimizing linear relaxation based guarantees can be helpful (Zhang et al., 2020). Despite the loose bound estimations and training complications, model training based on IBP often yields better certified test accuracy than more sophisticated approaches that optimize tighter bounds (Jovanović et al., 2022b).

Another noteworthy method for obtaining a form of robustness certificates is *randomized smoothing* (Lecuyer et al., 2019; Cohen et al., 2019). Here, a model is evaluated on many copies of input with random noise added to each one, and the average output is returned as the final prediction. Since randomized smoothing does not provide very strong bounds for l_∞ threat models, we do not make use of it in this dissertation.

While we focus on certifiable guarantees against adversarial perturbations, there are other types of provability in the context of image classification. PAC, learning theory On the other hand, it is in many cases not possible to provide certain types of correctness guarantees for the predictions of machine learning models. For example, as long as we do not obtain 100% accuracy on the test dataset of a certain distribution of images that fit a specified semantic description, we cannot even hope to guarantee that predictions on new, unseen inputs will be correct. Also for OOD detection, we cannot derive guarantees that no undetected OOD input exists, since we usually are not able to formally specify what potential OOD data can and cannot look like.

1.4 Outline

This dissertation is structured into three parts. The first part introduces the OOD detection problem, defines the objectives that are regarded when tackling this problem, and discusses how to evaluate the performance of a given OOD detector. For this, we initially discuss the definition of an individual input counting as OOD and the resulting correctness and coverage requirements for a good OOD test dataset. We illustrate the importance of clearly setting and ensuring these requirements by revealing how they are not satisfied by many evaluations that have been used in the OOD detection and OSR literature. Then we suggest a principled approach to building a dataset which does not have the aforementioned issues. We demonstrate the feasibility of constructing such a high quality OOD dataset by presenting NINCO, which we have crafted as a correct and challenging OOD dataset of natural images with respect to the ImageNet-1K in-distribution. We then provide an overview of post-hoc OOD detection methods, which are those which the community has focused on so far for ImageNet scale images, and thoroughly evaluate these methods on NINCO. Besides NINCO’s natural images, we also highlight the importance of detecting potential non-natural images like fully black, striped, or noisy images, as we argue that they might result from failures of cameras or image processing pipelines. We make the concerning observation that high confidence as well as failures of sophisticated purpose-built OOD detectors occur even on such supposedly easy to detect inputs, with e.g. completely black frames often being confidently identified as photos of match sticks.

In the second part, we analyze the basic building blocks of several established OOD detection methods. We mostly focus on methods that during training have access to both ID samples with their class labels and surrogate OOD data. While contrasting such surrogate OOD data with ID samples in a binary classification task yields an OOD detector that generalizes quite well to unseen out-distributions, we will see that it is beneficial to also use the class label information in some way or another when training a model used for OOD detection. While there are a few different ways to combine ID label and ID vs. OOD information in the training loss, it turns out that several methods minimize the same objective when we compare the Bayes optimal minimum of the function they aim to optimize. We further take a look at the binary discriminator that is equivalent using density estimation for OOD detection and argue that when expressed in this formulation, it is not surprising that density based methods have not produced convincing results on demanding OOD detection tasks.

In the third part, we introduce the issue of adversarial perturbations that can turn an OOD aware classifier’s low confidence predictions into very high confidence ones. After demonstrating that such perturbations are easily found for models not defended against these attacks, and showing that high confidence predictions are even inevitably assumed when moving very far away from the training data, we provide mitigations for this problem. First, we present ACET, a variant of adversarial training that defends against OOD overconfidence attacks, leading to empirically robust models. Then, in the last two chapters of the dissertation, with the GOOD and ProoD methods, we show that it is possible to construct models that are provably robust against confidence increasing perturbations on OOD data and also do not have

the problem of asymptotic overconfidence very far away from the training data. Remarkably, we observe that it is possible to achieve such robustness guarantees on OOD data without compromising the model's standard accuracy on ID test data.

Part I

**The Out-of-Distribution Detection
Problem**

Chapter 2

When is an Image Out-of-Distribution?

This chapter is based on the paper “In or Out? Fixing ImageNet Out-of-Distribution Detection Evaluation” (Bitterwolf et al., 2023) which was presented at ICML 2023. I shared first authorship with Maximilian Müller, we both worked on all aspects of the project and shared the work of curating the individual images used for evaluating and creating the OOD datasets. I first identified the extent of the issues in datasets used in previous works and the potential necessity of providing a new dataset devoid of those issues. Matthias Hein provided general guidance, valuable suggestions for aspects of both the analysis and the introduced datasets, and assistance with writing and revising the paper.

2.1 Requirements for an OOD dataset

In order to quantify the problems in a model’s behaviour when confronted with out-of-distribution classes, we need to compare its outputs on such OOD inputs with those on standard, in-distribution samples. The OOD detection metrics AUROC and FPR@95%TPR, which we defined in Section 1.2.1, measure the tendency of scores collected on ID test data being higher than scores on an OOD test dataset. In this chapter, we discuss what this required OOD test dataset can and should look like.

When we introduced the OOD detection problem in Section 1.2, we discussed the main motivation for the need of dedicated OOD detection solutions which is the issue of overconfidence of ID classifiers on OOD inputs. The corresponding baseline detector which tries to use the confidence as the score to judge whether an input is ID or OOD is in the following denoted as ‘Maximum Softmax Probability’, or short ‘MSP’.

A concrete MSP detector, which is obtained by fixing a confidence threshold, e.g. such that the TPR is 95%, counts it as a ‘fail to reject’ error against the model whenever it gives a confidence higher than this threshold in any ID class on an image marked as OOD. When, however, given an input image that really shows an object from an ID class, we usually are fine when the model predicts this class with high confidence, and often even *expect* the model to behave this way. This holds even if the images showing ID class objects are not sampled from the exact *in-distribution*. As a consequence, the standard OOD detection evaluations are not suited for detecting distribution shifts of input images that show an object from an ID class, e.g. due to background features, photography settings, regional characteristics etc.

that differ from the distribution represented by the training images. While detecting those as different can certainly be useful, especially since it often happens that misclassifications are more common on such inputs, it is not an error of the classifier if it confidently outputs the ID class that is shown in such images. Therefore, current OOD detection evaluation is restricted to the clear distinction where OOD test samples do not semantically belong to any of the ID classes.

To conduct these evaluations, we need one or ideally several datasets¹ of such OOD samples. The OOD datasets used for evaluation should meet the following criteria:

- They (at least together) should cover a **diverse range of semantic content** which includes inputs that can realistically be expected to be relevant. While we usually cannot assume that every possible type of unforeseen input can be thought of and collected for this evaluation, the more semantic space the dataset covers, the more likely it is that the evaluations reflect behaviour on truly unforeseen OOD inputs when the classifier or detector is deployed in the real world. For example, all kinds of animals from different regions might walk in front of the lenses of an autonomous vehicle, unknown food specialities might occur on holiday photos that are to be sorted automatically, or camera issues like optical failures or changed setups might lead to unrecognizable photos.
- They should contain **challenging** types of inputs. For example, the INATURALIST OOD PLANTS dataset described below contains a quite comprehensive range of plant photos which are (almost) all OOD, but as we will see below, most ImageNet models do not show problematic behaviour on this dataset, even if they turn out to do so on many other realistic OOD inputs. That is not to say that datasets like INATURALIST OOD PLANTS are not useful or should not be evaluated on, but they need to be accompanied by more challenging examples of OOD data in order to get a good picture of a models OOD problems and capabilities.
- Importantly, as discussed above, the images in an OOD test dataset need to **actually be OOD**. As we observe in detail for many datasets below, this requirement is not a trivial one to fulfill, particularly with automated data selection techniques. In the case of ImageNet-1K which we discuss in this section, there are 1000 concepts which need to be avoided. This makes it likely for randomly sourced images to contain such class content, and we found that indeed, the challenging ones of the datasets used for evaluating ImageNet OOD detection are indeed full of images that do contain objects from one or more of these classes. While even a large fraction of ID samples in a supposedly OOD dataset can often still lead to at least somewhat informative ranking of the quality of evaluated OOD detectors, we show examples where a better detector gets punished for indeed recognizing the ID class present in such false OOD images, we see

¹One single dataset often has either a limited range of semantic content or does not provide a useful structure which allows distinguishing performance on different facets of possibly occurring OOD inputs. This is why usually one wants to evaluate on a range of different OOD datasets.



Figure 2.1: **Contamination of OOD test sets with ID samples (ImageNet).** *Blue*: ImageNet-1K class found in the image. *(Brown)*: Label of the image in the original source dataset. **Top**: Samples from classes of the OOD dataset that by class meaning categorically overlap with ImageNet-1K classes. **Bottom**: Labels alone do not reveal that the images are ID, but incidental ID objects can be found.

large shifts in detection performance numbers between clean and contaminated OOD datasets.

- Further, we argue that an in-depth analysis of a models strengths and weaknesses when confronted with different types of OOD data can only be conducted if the data does not contain what it is not supposed to contain, and that, also with future, better detectors in mind, for good scientific measurements of OOD detection performance, one should not make compromises when it comes to the OOD dataset being truly OOD.

As we emphasize above, we strictly demand that an input image be considered OOD if and only if it does not contain objects from any of the in-distribution classes. We will show in the following that this assumption is not fulfilled for most of the test OOD datasets for ImageNet-1K (IN-1K) of Russakovsky et al. (2015) which existed until that point. We found the same issues with datasets from open set recognition (OSR) for IN-1K, which is evaluated on OOD

datasets which have the same requirements as those for OOD detection and thus would in principle, if they didn't have those issues, be valid OOD test datasets. We demonstrate that occurrences of objects from ID classes in test OOD datasets are often correctly recognized by state-of-the-art OOD detectors, but as an unwarranted consequence held against them as mistakes in OOD detection evaluations (false “false positive”). Even in cases where strong 2023 models struggle to identify ID content, e.g. if ID objects are partially occluded or in the background, OOD datasets containing ID objects are not future proof: when evaluating on them, one would not realize if a future model correctly predicts the class of a visible ID object.

The erroneous occurrences of ID objects in existing OOD datasets can be characterized into two failure modes, which we illustrate in Figure 2.1 and define as follows. **Categorical ID contaminations** show objects from ID classes which already are classes in a base dataset from which the test OOD dataset has been built. Their label coincides with an ID class or semantically designates a subset of an ID class, e.g. the class *hayfield* from the PLACES dataset and the IN-1K class *hay*. **Incidental ID contaminations** on the other hand occur in images which are supposed to belong to an OOD category but which contain an ID object. The object can be in the background or an aspect of the specific instance of the shown main object, e.g. the IN-1k class *plane* in an image of the OOD category *sky*. We show that ID contaminations strongly impact the conclusions which can be drawn from evaluating OOD detection methods by (1) systematically underestimating the true OOD detection performance and (2) unrightfully punishing stronger OOD detectors.

Compiling a test OOD dataset is indeed a challenging task, as the 1000 classes of IN-1K cover a fair portion of the images found in general image datasets. In Bitterwolf et al. (2023), we introduce the **NINCO (No ImageNet Class Objects) dataset** which contains 5 879 images that we individually checked not to contain any ID object from the classes of IN-1K. These images are ordered into 64 OOD classes, which facilitates a specific analysis of the failure modes of an OOD detector. Additionally, we provide a dataset of “**OOD unit-tests**”, synthetic images which do not resemble real world photos, but are designed to test specific weaknesses that might have impact in real-world applications (e.g. due to a camera failure). We find that surprisingly many OOD detectors struggle to detect these supposedly easy unit-tests, in particular methods that work well on natural test data.

Towards the end of this chapter, we provide a detailed OOD detection evaluation on NINCO for a range of eleven OOD detection methods across a large number of architectures and training schemes. Surprisingly, it turns out to be difficult for many OOD detectors to improve consistently over the baseline of Maximum Softmax Probability (MSP). While we confirm the observation that pretraining on larger datasets generally helps OOD detectors and particularly methods explicitly using pre-logit feature-information, we find that the type of pretraining has a strong impact.

2.2 Issues with previously existing OOD and OSR datasets for ImageNet-1K

We give an overview of the datasets that have been used to evaluate OOD detection performance for IN-1K as ID. In the following we use *blue* for the name of an ImageNet class and *brown* for the category name in the source dataset used for the generation of the test OOD dataset.

INATURALIST OOD PLANTS is a subset of 10 000 images curated by Huang & Li (2021) from 110 OOD plant species of iNat2017 (Van Horn et al., 2018) which is sourced from the iNaturalist project. It is frequently used as test OOD dataset (Xia & Bouganis, 2022; Ming et al., 2022).

PLACES is a subset of Places365 (Zhou et al., 2017) curated by Huang & Li (2021) as “50 categories [...] that are not present in IN-1K”. It is used as test OOD dataset in (Huang & Li, 2021; Sun et al., 2021; Ming et al., 2022). The dataset contains 9 822 images from 50 environment classes. We find that several of these classes are either subsets of ID classes, e.g. *hayfield* (*hay*), *cornfield* (*corn*), *lagoon* (*seashore* and *lakeshore*), or contain mostly ID objects, e.g. *underwater* (*coral reef* and *scuba diver*), *ocean* (*seashore*).

TEXTURES (Cimpoi et al., 2014) contains 5640 images of various objects that show one of 47 patterns. It is used as test OOD dataset in (Huang & Li, 2021; Sun et al., 2021; Wang et al., 2021; Xia & Bouganis, 2022; Ming et al., 2022) and others. Wang et al. (2022a) address the issue of overlap with IN-1K and remove four categorically ID textures (*bubbly* (*bubble*), *honeycombed* (*honeycomb*), *cobwebbed* (*spider web*), *spiralled* (*spiral*)). We find that even their version (denoted as TEXTURES43) contains about 20% ID images.

SPECIES was proposed in (Hendrycks et al., 2022) as OOD dataset for IN-21K (Deng et al., 2009) and should thus also be OOD for the IN-1K subset. Sourced from iNaturalist, it consists of 700 000 images from 1 316 species which were selected for not being in IN-21K. They sort the species into 10 superclasses. The largest superclass *Fungi* largely coincides with the IN-1K class *mushroom*, and also many of the remaining species are ID. Papers evaluating on SPECIES for IN-1K OOD detection include (Salehi et al., 2021; Yang et al., 2022; Song et al., 2022).

IMAGENET-O (Hendrycks et al., 2021) contains 2 000 images from IN-21K, excluding its subset IN-1K. To make the dataset challenging it was composed from images where a ResNet-50 classifier for a subset of 200 IN-1K classes attains high confidence. The samples being OOD relies on the assumption that IN-21K without IN-1K is OOD for IN-1K. However, this assumption does not hold, due to a significant overlap between ImageNet classes from IN-1K and IN-21K, e.g. *analytical balance/scale* and *pickle/cucumber*, and insufficient filtering for incidental ID objects.

OPENIMAGE-O (Wang et al., 2022a) consists of 17 632 images from the OpenImage-v3 (Krasin et al., 2017) test set which their human labellers categorize as OOD. It is also used in Yang et al. (2022).

360OPENSETCLASSES (Bendale & Boult, 2016a) uses those 360 classes (15.000 samples)

from ILSVRC2010 which are not part of ILSVRC2012. Like for IMAGENET-O, this leads to large semantic overlap, e.g. the class *organ pipe* coinciding with the ID class *organ*.

SEMANTIC SHIFT BENCHMARK (SSB) (Vaze et al., 2022) contains a *hard* and *easy* OSR benchmark, each consisting of 1000 classes, that were created by regarding the distances between nodes in the WordNet tree. Similar to 360OPENSETCLASSES, we find both categorical and incidental ID contamination, e.g. *rainbow lorikeet/lorikeet*. Papers evaluating on SSB include (Wen et al., 2023).

IMAGENET-1K-OOD (Wang et al., 2022b) contains 50.000 images from 1.000 classes randomly sampled from ImageNet-21K, such that those classes don't overlap with ImageNet-1K and ImageNet-LT, another dataset introduced by the authors. Categorical examples include *bobwhite quail/quail* and *king vulture/vulture*.

COOD-BENCHMARK (Galil et al., 2023) is a general framework for benchmarking ImageNet-1K OOD detection. Their test set consists of ImageNet-21K samples which were filtered by class. It includes severe contamination, including categorical cases like *orange, orange tree/orange*.

2.2.1 Prevalence of ID samples in popular OOD datasets

Concerningly, several test OOD datasets for IN-1K that have been in use by the community contain a substantial fraction of samples that show ID objects. Figure 2.1 shows some typical appearances of ID data in supposedly OOD datasets. The categorical ID failure mode illustrated in the top part is the inclusion of samples from explicitly ID classes of the source dataset from which the OOD dataset has been built. For instance, the class *hayfield* from the PLACES-dataset overlaps with the IN-1K class *hay*. However, also in principally innocuous classes (bottom part), many incidental ID samples can still be found. Here, the occurring failure modes are numerous: some ID objects happen to be in the background, some are a prominent part of the depicted scene, and some happen to realize both the original class and the ID class. For instance, the class *table knife* contains samples which also show a *plate*, and the class *striped* from the TEXTURES-dataset often shows the stripes of a *zebra*.

In order to quantify the severity of ID objects in test OOD datasets, we manually check for ID objects in 400 random samples from each of the most commonly used datasets. For fair treatment, unclear and ambiguous samples, which we would exclude from NINCO introduced below, are ignored in this survey. The results in Table 2.1 show that for many of these common OOD detection benchmarks, a substantial fraction of samples is actually ID: For both the PLACES and SPECIES datasets, it is more than 50%. Only INATURALIST OOD PLANTS (2.5% of samples ID) and OPENIMAGE-O (4.9% ID) contain comparably few ID images.

2.2.2 Effect of ID contamination on OOD evaluation

In Figure 2.2, we show how OOD detection evaluation with incidental ID samples can unrightfully punish strong OOD-detectors: A better model can correctly recognize ID objects with



Figure 2.2: A **Vision Transformer** confidently classifies ID objects in samples from popular OOD datasets (*source label in parentheses*) as the correct IN-1K class, but is marked down with false positives in OOD detection evaluation when using MSP (Max Softmax Prob.) as criterion. The weaker **ResNet-50**, in contrast, doesn’t recognize the ID objects and hence the MSP is low enough to reject all images wrongly as OOD. This illustrates how **a better model (ViT in our case) can be unjustly punished when the test OOD dataset contains ID objects**. For both models, the 95%TPR threshold is at a MSP of 38%. Origins of the images: PL=PLACES, SP=SPECIES, OO=OPENIMAGE-O, IO=IMAGENET-O.

high confidence even if they are in the background of the image, leading to a false “false positive” in the evaluation, while a weaker model not recognizing the ID object and providing a low-confidence prediction is “rewarded” with a false “true negative”. For example, the strong VisionTransformer (ViT) (Dosovitskiy et al., 2021) identifies the *pole* besides an otherwise empty desert road, and thus has high confidence on the image where the weaker ResNet-50 does not recognize any ID class with high confidence. Similarly, in the second example, the ViT is punished with a false “false positive” for recognizing (above the detection threshold) the *oranges* in the background while ignoring the unknown flying fox (truly OOD), whereas the ResNet-50 even does predict a wrong ID class, namely *squirrel monkey*, but does so with low confidence (below the detection threshold), and is thus rewarded with a false “true negative”.

We quantify the effect of ID contaminations on evaluation results in customary OOD datasets in Figure 2.3 for the MSP baseline and the Mahalanobis OOD detection method (Lee et al., 2018b). For the test OOD datasets which showed a large portion of ID samples in Table 2.1, we report the FPR at 95% TPR obtained with a ViT when evaluating on the original

Table 2.1: **Percentage of ID samples, $p = \frac{\text{ID}}{\text{ID}+\text{OOD}}$, in commonly used test OOD datasets** found by visual inspection of 400 random samples per dataset. Unclear samples are ignored (which are at most 6.7% (for PLACES) of the 400 samples).

Dataset	ID samples	Dataset	ID samples
PLACES	59.5%	SPECIES	57.0%
IMAGENET-O	20.2%	TEXTURES	25.6%
INAT. PLANTS	2.5%	TEXTURES43	20.0%
OPENIM.-O	4.9%	IN-1K-OOD	32.1%
SSB-HARD	41.6%	SSB-EASY	53.4%
360OPENSET	26.9%	COOD	38.2%

400 samples and our cleaned subsample of it not containing any more ID objects. We find that ID contaminations strongly impact the conclusions which can be drawn from evaluating OOD detection methods on those datasets. Most clearly, both methods perform substantially *better* after removing the images with ID objects from the OOD datasets, in some cases reducing the FPR by more than 50%. This is unsurprising: If a significant fraction of the dataset is actually ID, this fraction should not be detected as OOD by a well-performing method. Hence, evaluating OOD detection performance with partially ID data leads to a systematic *overestimation* of the true FPR of the OOD detection method and disadvantages better models as they are more likely to detect ID objects as discussed above. Additionally, we observe that the differences between OOD detectors become more pronounced. In Figure 2.3 it can be seen that for each dataset, the FPR for the Mahalanobis OOD detector decreases more than for the MSP-baseline. The effect is particularly strong for SPECIES (25.6% gain of MSP vs. 33.2% gain of Mahalanobis) and PLACES (19.6% gain vs. 26.3% gain), which are the two datasets we found to contain most ID samples. We further emphasize that due to the presence of large fractions of ID samples in most common benchmarks, even the performance of a perfect detector would saturate significantly above 0% FPR. For example with SPECIES, we find that for a strong current detector already more than 85% of the 'false positives' contain ID objects.

2.3 NINCO: A new OOD test set for ImageNet-1K

As discussed in Sec. 2.1, an **OOD input for IN-1K** is an image that does not contain an object from one (or several) of the 1 000 IN-1K classes. These ImageNet classes are based on individual WordNet (Fellbaum, 1998) synsets, each consisting of one or more keywords that are synonymous in some context. During the ImageNet creation process (Deng et al., 2009), images were first collected from the web by using variations of each keyword of a respective class and then verified by humans to fit its synset's definition.

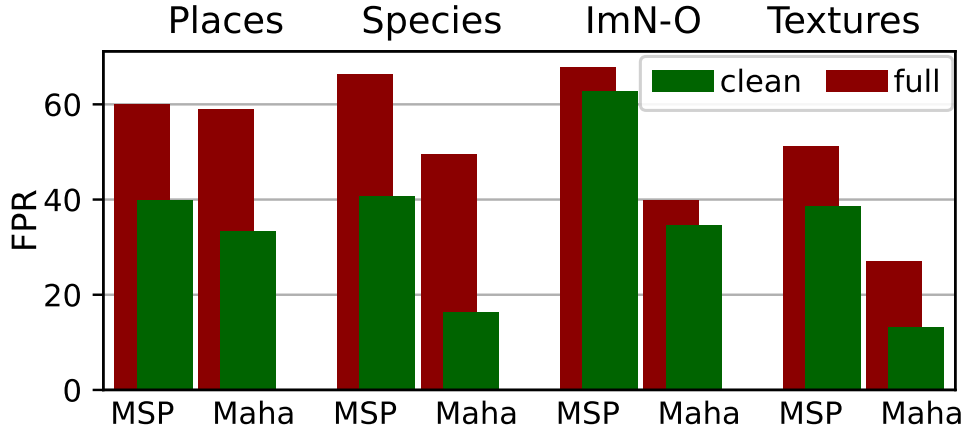


Figure 2.3: **OOD-detection before and after removing samples with ID-objects:** We show FPR (lower is better) of two OOD detectors (MSP and Mahalanobis distance) for a ViT, evaluated on cleaned and full subsets of four popular OOD datasets.

Sourcing OOD test samples for ImageNet-1K from ImageNet-21K (or its subsets) based on class-labels has been leading to highly contaminated datasets (5 of the datasets in Table 2.1 are sourced from ImageNet-21K and all contain between 20% and 53% ID samples and show significant categorical contamination). This is partly due to the class-structure of those datasets: Both ImageNet-1K and ImageNet-21K contain leaf and internal nodes of the WordNet-tree as classes. While the internal nodes of ImageNet-1K are not ancestors to other ImageNet-1K classes, ImageNet-21K internal nodes can be ancestors to ImageNet-1K nodes, and vice versa. Moreover, there are ambiguous class-definitions in WordNet, like e.g. *police dog*, which is not parent or child of another dog class, but mostly shows a *german shepherd*, or an *alley cat* showing one of the many cat classes without being parent or child to other cat classes. Besides, there is significant incidental contamination even for nominally disjoint classes. Since the automation of filtering for challenging OOD data would require a strong detector that already solves the problems that the dataset is meant to pose, we conclude that it is impossible to construct a clean and challenging OOD dataset without manually checking the OOD samples for ID contamination.

In reality, many ImageNet samples fit one but not necessarily *all* keywords of their class label. This means that to make sure that OOD detectors are treated fairly², OOD test samples cannot fall into the definition of any keyword of any IN-1K class. For example, photos of the Sumatran orangutan cannot be considered OOD, since they could be included in the IN-1K class (*orangutan*, *orang*, *orangutang*, *Pongo pygmaeus*), even though *Pongo pygmaeus* only refers to the Bornean orangutan. To determine what counts as an ID object, we follow the WordNet glosses, which for each synset can be looked up at <http://wordnetweb.princeton.edu/perl/webwn>, as well as dictionary definitions of

²For fair treatment of previous OOD *datasets*, such unclear samples that don't fit all keywords were ignored in Table 2.1.



Figure 2.4: **Difficult OOD classes in NINCO**: Examples of images from some of NINCO’s most difficult (see Tables 2.3/ 2.4) *OOD classes* (first row) and from the *ImageNet-1K class* (second row) which the *OOD class* is most frequently confused for.

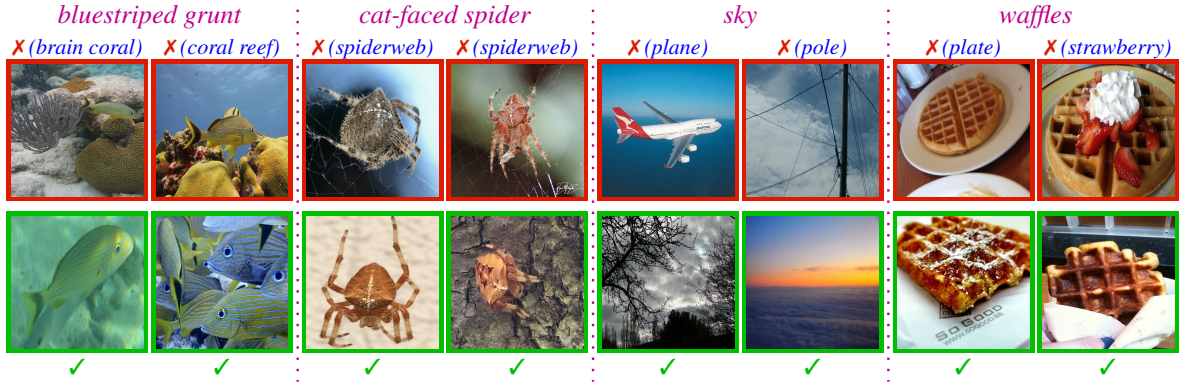


Figure 2.5: **Cleaning the *OOD classes* that eventually constitute NINCO**. **Top**: Samples that were excluded due to overlap with ID classes, which are noted in *blue*. **Bottom**: Samples from the same OOD class that are included in the cleaned NINCO dataset.

keywords and source dataset class labels. For difficult cases, we consult additional sources like Wikipedia. For example, the species *northern elephant seal* does not fall into the ID class *sea lion*, among other biological criteria distinguished by the fact that the former do not have ears while the latter do. An image of an OOD dataset can furthermore not incidentally contain ID objects, to avoid cases as in Figure 2.1 (bottom) and Figure 2.2.

2.3.1 NINCO dataset construction

For each OOD class of our new NINCO dataset, we start by **choosing a base class** which consists of all samples from a named class of an existing or newly scraped dataset. The majority of the NINCO base classes are sourced from SPECIES (Hendrycks et al., 2022), which provides images scraped from iNaturalist. For each base class, we carefully decide, based on

WordNet glosses, iNaturalist taxonomy details and Wikipedia, whether it can be included according to the non-permissive interpretation described at the beginning of Section 2.3. The choice of base classes is not random, since there is no way to randomly sample from the set of concepts that might occur at test time. Rather, we aim for a variety of classes that are challenging, diverse and, most importantly, not actually categorically ID to begin with. Then for each base class, we **individually inspect each image** for ID objects. To help remembering the 1000 ID classes, we display the 5 top ID classes of a ViT’s prediction on each image. If an ID object is at least partially visible, the corresponding sample is removed. In cases where it is ambiguous whether we see an ID object in the image, the sample is not included in the cleaned dataset. As the iNaturalist data (including the SPECIES dataset) has been curated by experts and can be considered very reliable, we generally trust in the main object belonging to the species it is labelled as. For base classes chosen from the other sources, we consider ourselves competent to verify whether a label is correct. In addition to samples showing ID objects, we also remove images where no object from the OOD class is visible, e.g. we exclude pictures of animal traces or remains which frequently appear in iNaturalist. While for most existing datasets, the cleaning has been outsourced to external services like Amazon Mechanical Turk or student labellers. By researching all OOD classes and visually inspecting all their samples ourselves, we as authors of NINCO were able to do more in-depth research for each ambiguous case and obtain more coherent decisions, which we are positive leads to a higher quality dataset. Such high data quality is crucial for in-depth evaluations (Vasudevan et al., 2022; Shankar et al., 2020), as only being completely in-distribution free allows understanding a detector’s individual mistakes.

The NINCO (No ImageNet Class Objects) dataset consists of 64 OOD classes with a total of 5 879 samples. The base classes which we cleaned to obtain NINCO were sourced from SPECIES (35 classes) (Hendrycks et al., 2022), PLACES (3 classes) (Zhou et al., 2017), which both are discussed in Section 2.2, as well as from the FOOD-101 dataset (7 classes) (Bossard et al., 2014), CALTECH-101 (4 classes) (Li et al., 2022), MYNURSINGHOME (4 classes) (Ismail et al., 2020), ImageNet-21k (1 class) and newly scraped from iNaturalist.org (2 classes) or other websites like Flickr (8 classes). For details and samples for all NINCO OOD classes please refer to the published paper (Bitterwolf et al., 2023). In addition to NINCO, we also provide the 2715 OOD images obtained from cleaning 400 samples of eleven test OOD datasets as discussed in Section 2.2.2. In order to notice ID contaminations potentially biasing the drawn conclusions, we recommend to also evaluate on these cleaned versions when evaluating on those original benchmarks.

2.3.2 OOD unit-tests

Following common practice (e.g. Hendrycks et al. (2022)), we argue that evaluating an OOD detector on a range of simple, synthetic classes *besides* the variably challenging natural image classes of an OOD dataset can give additional insights about its OOD detection weaknesses. Since these **OOD unit-tests** do not represent a diverse distribution of photos, but different modes of simple, synthetically generated image inputs which any good OOD detector should

be expected to detect, we don't include them in summary metrics or distribution plots. Instead, we suggest to count an OOD unit test as **failed** if a method has an FPR above a user-defined threshold, which we suggest setting at 10%, and to report the number of *failed* OOD unit-tests (which should be 0 for a strong OOD detector) alongside the aggregate results on a test OOD dataset like NINCO. For each OOD unit-test, we provide a set of 400 samples in typical ImageNet format, by mirroring the sizes and file formats of random ImageNet samples. While some OOD unit-tests may appear redundant at first sight, we find that they provide important information as some detectors e.g. mostly pass the *monochrome* test but completely fail on *black*, which reveals a specific weakness that is very realistic to be encountered in practice.

2.4 OOD detectors for ImageNet and how to evaluate them

An **OOD detector** for inputs from the domain X of possible input images is represented by a score function $S : X \rightarrow \mathbb{R} \cup \{\pm\infty\}$ which is generally supposed to be larger on ID inputs than on OOD inputs. One example is the Maximum Softmax Probability (MSP) or confidence $S_{\text{MSP}}(x) = \max_{k=1,\dots,K} p_k(x)$ of a classifier with output probabilities p for K ID classes. The MSP is the standard baseline OOD detection method (Hendrycks & Gimpel, 2017), since it is intuitively expected to be low on OOD compared to ID inputs. Observing that standard classifiers are frequently overconfident on OOD inputs, OOD detection research aims at finding detectors that improve on this baseline.

2.4.1 OOD detection methods

We give an overview of a range of OOD detection methods which have been proposed for IN-1K as ID. For clarity, we denote vectors in bold and lowercase letters and matrices in bold and uppercase letters. We write neural networks as functions n , which are parametrized by weights θ , take an input sample $\mathbf{x}$ and produce an output vector $\mathbf{o}$ of size C , where C is typically the number of classes in a classification task (1000 in the case of IN-1K). We refer to $\mathbf{o}$ as the logits of $\mathbf{x}$, which can be transformed to a probability vector $\mathbf{p}$ (also of size C) via the softmax function: $p_i = \exp(o_i) / \sum_c \exp(o_c)$. The network n can be decomposed into a feature extractor h and the networks last layer g :

$$\mathbf{o} = n(\mathbf{x}) = g(h(\mathbf{x})),$$

where g is a fully connected, linear layer, i.e. $g(\mathbf{h}) = \mathbf{W}^T \mathbf{h} + \mathbf{b}$ with weight $\mathbf{W}$ and bias $\mathbf{b}$. We refer to $\mathbf{h} = h(\mathbf{x})$ as the *features* or the *embeddings* of $\mathbf{x}$ w.r.t. the network n . As presented in Section 2.5, for each sample $\mathbf{x}$, a method returns an OOD-score $s = f(\mathbf{x})$, a scalar value which is supposed to be larger for ID data and smaller for OOD data. Methods accessing $h(\mathbf{x})$ directly in order to compute the OOD-score are referred to as feature-based methods, in contrast to methods that derive their OOD-score from the logits $\mathbf{o}$ (even though obviously the logits implicitly also depend on these features). In the following, we will describe how each method computes the score s for a test input $\mathbf{x}$.

MSP (Hendrycks & Gimpel, 2017): The standard OOD-detection baseline uses the confidence, i.e. the max softmax probability of a models probability output vector:

$$s = \max_c(p_c)$$

Max-Logit (Hendrycks et al., 2022): Similar to MSP, Max-Logit returns the largest entry of the logit-vector $\mathbf{o}$, i.e.

$$s = \max_c(o_c)$$

Energy (Liu et al., 2020): The Energy based OOD detection method uses the denominator of the softmax-function as OOD-score:

$$s = \log \sum_c \exp(o_c)$$

KL-Matching (Hendrycks et al., 2022): KL-Matching computes a mean probability vector $\mathbf{d}_c$ for each of the C classes. For a test input, the KL-distances of all $\mathbf{d}_c$ vectors to its probability vector $\mathbf{p}$ are computed, and the OOD-score is the negative of the smallest of those distances:

$$s = -\min_c \text{KL}[\mathbf{p}||\mathbf{d}_c]$$

In the original paper by (Hendrycks et al., 2022), the average for $\mathbf{d}_c$ is computed over an additional validation set. Since none of the other methods leverages extra data and we are interested in fair comparison, we deploy KL-Matching like in (Wang et al., 2022a; Yang et al., 2022), where the average is computed over the train set.

KNN (Sun et al., 2022): KNN is a non-parametric method that computes distances in the feature-space. Specifically, the feature vector of a test input is normalized to $\mathbf{z} = \mathbf{h}/\|\mathbf{h}\|_2$ and the pairwise distances $r_i(\mathbf{z}) = \|\mathbf{z} - \mathbf{z}_i\|_2$ to the normalized features $\mathcal{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$ of all samples of the training set are computed. The distances $r_i(\mathbf{z})$ are then sorted according to their magnitude and the K^{th} smallest distance, denoted $r^K(\mathbf{z})$ is used as negative OOD-score:

$$s = -r^K(\mathbf{z})$$

Like suggested in (Sun et al., 2022), we use $K = 1000$.

Mahalanobis distance (Lee et al., 2018b): This popular method fits a class-conditional Gaussian with shared covariance matrix to the train set, i.e. computes

$$\hat{\mu}_c = \frac{1}{N_c} \sum_{i:y_i=c} \mathbf{h}_i, \quad \hat{\Sigma} = \frac{1}{N} \sum_c \sum_{i:y_i=c} (\mathbf{h}_i - \hat{\mu}_c)(\mathbf{h}_i - \hat{\mu}_c)^T$$

where N_c is the number of train samples in class c and N is the total number of train samples. The OOD-score of a test sample is then the Mahalanobis distance induced by $\hat{\Sigma}$ between its

feature $\mathbf{h}$ and the closest class mean:

$$s = -\min_c (\mathbf{h} - \hat{\mu}_c) \hat{\Sigma}^{-1} (\mathbf{h} - \hat{\mu}_c)^T$$

Relative Mahalanobis distance (Ren et al., 2021): A modification of the Mahalanobis distance method, thought to improve near-OOD detection, is to additionally fit a global Gaussian distribution to the train set without taking class-information into account:

$$\hat{\mu}_{\text{global}} = \frac{1}{N} \sum_i \mathbf{h}_i, \quad \hat{\Sigma}_{\text{global}} = \frac{1}{N} \sum_i (\mathbf{h}_i - \hat{\mu}_{\text{global}})(\mathbf{h}_i - \hat{\mu}_{\text{global}})^T$$

The OOD-score is then defined as the difference between the original Mahalanobis distance and the Mahalanobis distance w.r.t. the global Gaussian distribution:

$$s = -\min_c \left((\mathbf{h} - \hat{\mu}_c) \hat{\Sigma}^{-1} (\mathbf{h} - \hat{\mu}_c)^T - (\mathbf{h} - \hat{\mu}_{\text{global}}) \hat{\Sigma}_{\text{global}}^{-1} (\mathbf{h} - \hat{\mu}_{\text{global}})^T \right)$$

ReAct (Sun et al., 2021): The authors propose to perform a truncation of the feature vector, $\bar{\mathbf{h}} = \min(\mathbf{h}, r)$, where the min operation is to be understood element-wise and r is the truncation threshold. The truncated features can then be converted to so-called rectified logits via $\bar{\mathbf{o}} = g(\bar{\mathbf{h}}) = \mathbf{W}^T \bar{\mathbf{h}} + \mathbf{b}$. While the rectified logits can now be used with a variety of existing detection methods, we follow (Sun et al., 2021) and use the rectified Energy as OOD-score:

$$s = \log \sum_c \exp(\bar{o}_c)$$

As suggested in (Wang et al., 2022a), we set the threshold r such that 1% of the activations from the train set would be truncated.

Virtual Logit Matching (Wang et al., 2022a): The idea behind ViM is that meaningful features are thought to lie in a low-dimensional manifold, called the principal space P , whereas features from OOD-samples should also lie in $P^\perp$, the space orthogonal to P . P is the D -dimensional subspace spanned by the eigenvectors with the largest D eigenvalues of the matrix $\mathbf{F}^T \mathbf{F}$, where $\mathbf{F}$ is the matrix of all train features offsetted by $\mathbf{u} = -(\mathbf{W}^T)^+ \mathbf{b}$ (+ denotes the Moore-Penrose inverse). A sample with feature vector $\mathbf{h}$ is then also offset to $\tilde{\mathbf{h}} = \mathbf{h} - \mathbf{u}$ and can be decomposed into $\tilde{\mathbf{h}} = \tilde{\mathbf{h}}^P + \tilde{\mathbf{h}}^{P^\perp}$, and $\tilde{\mathbf{h}}^{P^\perp}$ is referred to as the *Residual* of $\mathbf{h}$. ViM leverages the Residual and converts it to a virtual logit $o_0 = \alpha \|\tilde{\mathbf{h}}^{P^\perp}\|_2$, where

$$\alpha = \frac{\sum_{i=1}^N \max_c o_i^c}{\sum_{i=1}^N \|\mathbf{h}_i^{P^\perp}\|_2}$$

is designed to match the scale of the virtual logit to the scale of the real train logits. The virtual logit is then appended to the original logits of the test sample, i.e. to $\mathbf{o}$, and a new probability

vector is computed via the softmax function. The probability corresponding to the virtual logit is then the final OOD-score:

$$s = -\frac{\exp(o_0)}{\sum_{c=1}^C \exp(o_c) + \exp(o_0)}$$

Like suggested in (Wang et al., 2022a), we use $D = 1000$ if the dimensionality of the feature space d is $d \geq 2048$, $D = 512$ if $2048 \geq d \geq 768$, and $D = d/2$ rounded to integers otherwise.

Cosine (Tack et al., 2020; Galil et al., 2023): This method computes the maximum cosine-similarity between the features of a test-sample and embedding vectors $\tilde{\mathbf{u}}_c$ (sometimes also called concept-vector):

$$s = \max_c \tilde{\mathbf{u}}_c^T \mathbf{h} / \|\tilde{\mathbf{u}}_c\|_2 \quad (2.1)$$

For zero-shot CLIP, $\tilde{\mathbf{u}}_c$ can be obtained by creating text-embeddings from the ImageNet class names. Encoding ‘A photo of a ...’ yields an embedding from the corresponding class. For classifiers, we use the class-wise train means $\hat{\mu}_c$, that are also used for Mahalanobis distance.

MCM/RCos (Ming et al., 2022; Techapanurak et al., 2020): Maximum-Concept-Matching was recently introduced as a zero-shot OOD detection method for CLIP and applies additional softmax-scaling to the cosine-similarities of the *Cosine* method, potentially with a temperature scaling (which we omit, following (Ming et al., 2022)). Again, we extend this method to work with conventional classifiers by using the class-means $\hat{\mu}_c$ like they are used for Mahalanobis distance as embedding/concept vectors. We then refer to it as relative cosine (short: MCM/RCos or just RCos) in order to distinguish it from CLIPs zero-shot method.

2.4.2 Evaluating OOD detectors on NINCO

An OOD detector is usually obtained by combining such an OOD detection method with a concrete classifier model. We analyze OOD detectors in terms of the fraction of falsely accepted OOD inputs at a true positive rate of 95%, short FPR.

Different OOD classes (and similarly also different test OOD datasets) represent different probabilistic distributions of inputs that a detector is tested against. An important arising question is how the collective of individual performance measurements can be interpreted and whether they can be aggregated into one number that can be used to make an informed decision on which OOD detector works best. Certainly, the notion of ‘best’ may notably vary depending on the application and situation and we often cannot hope to model a ‘true’ out-distribution, or even be sure that it meaningfully exists. An aggregate number which gives a good overview of an OOD detector’s performance on the class based NINCO dataset is the **mean FPR** of the individual FPR values for each of the 64 OOD classes of NINCO.

However, for many applications it is not possible to model the potential OOD inputs that might be encountered at test time with a fixed probability distribution. Thus a single aggregate number cannot tell the full story, and may hide outliers in the FPR values. For one, some errors might be *less acceptable* than others, e.g. a FPR of 20.0% might be very bad for monochrome

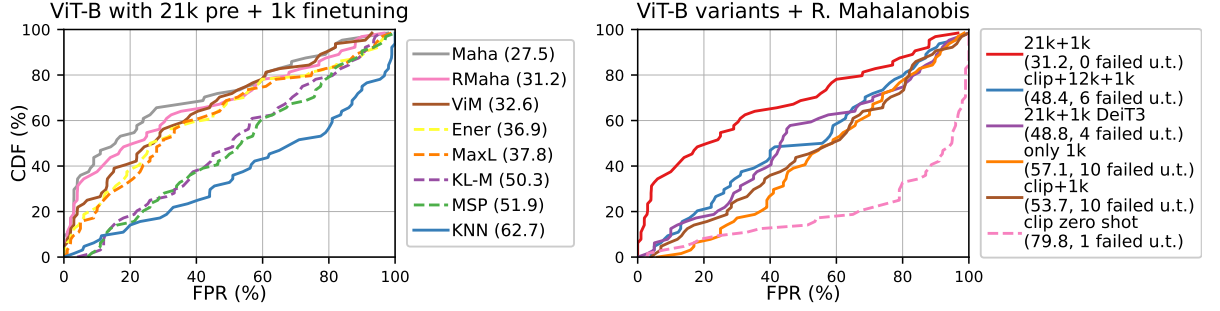


Figure 2.6: Cumulative distribution of the % of NINCO-classes for which an FPR at least as low as a given x-value is achieved. The area over this curve corresponds to the mean FPR. The further in the top left corner, the better. **The best methods explicitly access pre-logit features (Left):** Different OOD detection methods with a ViT-B pretrained on IN-21k (mean FPR in parentheses, pre-logit feature-accessing methods are solid, others dashed). **Not all pretraining helps (Right):** RMaha applied to ViT-B with different training variants (MCM for CLIP zero-shot is dashed). Only the top model does not fail OOD unit-tests.

inputs, but would lose much significance when subsumed into a mean. For OOD unit tests, where OOD detectors can be expected to be very robust, we therefore propose regarding pass-fail statistics instead of mean FPR. Also, an evaluator might want to be informed about the *concrete failure modes* of the model, e.g. all OOD classes with a particular high FPR. An OOD detector showing consistent improvements on most of the OOD classes (instead of only in terms of the mean) can be seen as strong evidence for the method yielding actual improvement, as opposed to the detector overfitting to a limited scope of test OOD data, which Wang et al. (2022a) describe as a form of hackability. Due to these considerations, and with the OOD data being organized into *OOD classes* as in NINCO, we suggest evaluations of OOD detectors to always provide the **distribution of results over OOD classes** and additionally to **make the individual results available**, such that the reader can make an informed comparison based on which types of OOD inputs are most relevant to them.

2.5 Evaluation results for IN-1K OOD Detectors

We evaluate a range of IN-1K models obtained from the public timm-library (Wightman, 2019) and state-of-the-art OOD-detection methods on NINCO. We focus on transformer architectures and convolutional networks, both with and without pretraining. While most pre-trained models were initially trained on IN-21K, we also include an EfficientNet trained via noisy student (Xie et al., 2020) on the JFT-300M dataset, and four ViTs with CLIP-pretraining (Radford et al., 2021) and subsequent fine-tuning, as well as a zero-shot CLIP model.

The ImageNet-1K OOD detection methods which we evaluate on NINCO can be grouped into two categories: Max-Softmax (MSP) (Hendrycks & Gimpel, 2017), Max-Logit (Hendrycks et al., 2022), Energy (Liu et al., 2020) and KL-Matching (Hendrycks et al.,

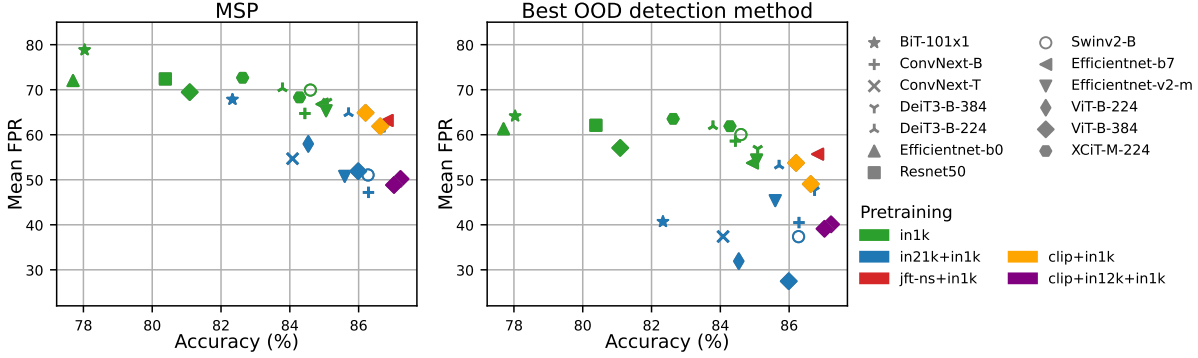


Figure 2.7: **IN-21K pretraining boosts feature-based OOD detectors on NINCO:** Mean FPR vs. accuracy for MSP and each model’s best detector, which (except for the noisy-student model) always explicitly accesses the pre-logit features. OOD detection strongly improves when using models pre-trained on IN-21K. Additional CLIP-pretraining or on JFT can yield higher accuracy, but OOD detection need not be better than with IN-21K pretraining.

2022) derive an OOD-score exclusively from logit outputs, whereas Mahalanobis distance (Maha) (Lee et al., 2018b), Virtual Logit Matching (ViM) (Wang et al., 2022a), ReAct (Sun et al., 2021), Relative Mahalanobis distance (RMaha) (Ren et al., 2021), and K-Nearest-Neighbours (KNN) (Sun et al., 2022) also leverage explicit information from the features of the DNN’s penultimate (pre-logit) layer. For the zero-shot evaluation of CLIP, we use Maximum-Concept-Matching (MCM) (Ming et al., 2022) and Cosine-similarity (Cos) (Galil et al., 2023) to class-specific text-embeddings. Noting that OOD detection based on softmax of a cosine similarity to a specific feature vector has been proposed in different variants (Tack et al. (2020), Techapanurak et al. (2020) and MCM); we find that using it with classifier class means produces reasonable OOD detection results, marked as relative cosine class similarity (RCos). We call those methods which explicitly access the pre-logit feature layer *feature-based*.

2.5.1 Results on NINCO

Comparison of OOD detection Methods. In Figure 2.6 (left), we tratre the performance of a single ViT when combined with a range of OOD-methods. Overall, most feature-based methods, like Maha, RMaha and ViM, outperform the MSP-baseline by a clear margin. Notably, MaxLogit and Energy, which do not explicitly access the pre-logit features, are also able to strongly improve over MSP, while KL-Matching performs roughly on par, and KNN much worse. We observe that while Maha, RMaha and ViM improve over MSP in all FPR ranges, this is different for e.g. MaxLogit: For large FPR, it is similar to MSP, indicating that the method brings no advantage over MSP for hard test classes, and its improved mean performance is mainly due to lower FPR for the easier OOD classes. When regarding the mean FPR values of all method-model-combinations shown in Tables 2.3&2.4, we observe

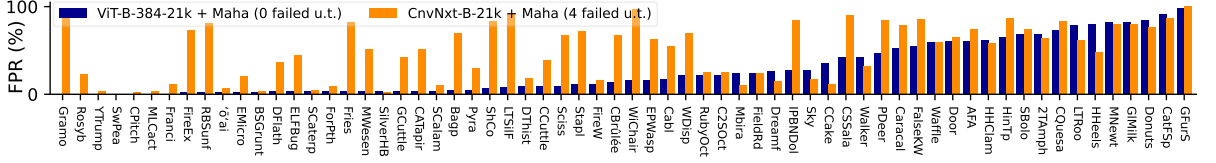


Figure 2.8: FPR of a pretrained ViT-B and pretrained ConvNext-B for all classes of NINCO.

that while Maha in combination with a (pretrained) ViT is the single best OOD-detector, this method often performs worse when combined with other models. RMaha, however, yields good results with *all* models, and is together with (Relative) Cosine the only method which can fairly consistently improve over the MSP baseline in terms of mean FPR. For most models, it is either the best-performing method, or close to the best-performing method, which is somewhat surprising, given its relatively poor performance on the unit-tests. We further note that for all models (except the noisy-student model), the best-performing method always explicitly accesses the pre-logit features, and that in contrast to e.g. KNN, Energy and ReAct, even the adapted methods based on feature space cosine similarity Cos and MCM/RCos fairly consistently improve over the MSP-baseline.

Pretraining matters. In Figure 2.7, we plot the mean FPR on NINCO over the accuracy for all investigated models for both the MSP-baseline (left) and the best-performing OOD detector per model (right). For MSP, the mean FPR decreases roughly linearly with accuracy. Since most pretrained models (blue) have higher accuracy, they typically also show better OOD-detection performance, but also between models of similar accuracy, the pretrained ones achieve better mean FPR. For the best-performing OOD detector, improvements can be observed for models both with and without pretraining. Notably, the linear relation between FPR and accuracy disappears, and all purely 1K models (green) perform roughly on one level. In comparison, the gains for the majority of models pretrained on IN-21K (blue) are larger. In particular ViT and BiT benefit strongly from leveraging their respective best method, which as discussed above is always feature-based. In other words, pretraining helps in two ways: First, it leads to higher ID-performance (accuracy), which benefits methods like the MSP-baseline. Second, it creates better feature-embeddings for this task, which lead to improvements beyond the accuracy-MSP correlation. This is most clearly visible for the pretrained BiT-m, which has comparably low accuracy (82%) and hence no outstanding MSP-performance, but outperforms all 1k-models by a significant margin with features leveraging ViM. However, as we observe in Figure 2.6 (right), the benefit of pretraining depends strongly on the specific data and training method: With RMaha, the ViT with ‘traditional’ IN-21K pretraining from (Steiner et al., 2022) clearly outperforms models with the distillation-based training of DeiT3 (Touvron et al., 2022), CLIP-pretraining or even CLIP with interjected IN-12K training. The zero-shot methods for CLIP, despite having shown promising results in (Galil et al., 2023) and (Ming et al., 2022) and performing well on the unit tests, are not competitive to IN-1k classifiers on NINCO. Regarding all methods, the five models trained with different pretraining strategies (EfficientNet-b7 with noisy student and four ViTs with CLIP-pretraining (Radford

et al., 2021) and subsequent fine-tuning) show some of the highest accuracies in our survey, yet, their OOD-detection performance is surprisingly poor. Overall, we see strong indication that the precise type of pretraining has a large impact on whether it produces a feature space that is beneficial for feature based methods.

Analysis of failure cases. In Figure 2.8 we plot the individual FPR for each OOD class of NINCO for the combination ViT+Maha, the overall best OOD detector in terms of mean FPR, and contrast it with ConvNext+Maha, which also shows good mean FPR. Performance varies widely between OOD classes, with both models severely struggling for some classes. Where the ViT shows large FPR, the ConvNext rarely performs better, while it also fails to detect certain classes like the *long-tailed silverfish* where the ViT does well. We illustrate samples from hard classes in Figure 2.4. Both models struggle to detect the *Galápagos fur seal* (98% FPR for the ViT), often confused with the IN-1K class *sea lion*, and *cat-faced spider* (confused with *barn spider*, 91% FPR). From a human perspective, those classes are arguably hard to detect. We note, however, that it is possible to tell them apart, as a ViT IN-21K-classifier e.g. identifies the *Galápagos fur seal* as a *fur seal* (IN-21K class) in 92% of samples and misclassifies only 6% of them as a *sea lion*. The networks however also fail for classes more obvious to humans: *donut* (84% FPR ViT, confused with *bagel*), *spaghetti bolognese* (69% FPR, *carbonara*) and *chicken quesadilla* (73% FPR, *burrito*) also confuse both models.

2.5.2 Results on the OOD unit-tests

Auditing OOD detectors on the OOD unit-tests, we find that surprisingly many combinations of models and OOD detection methods struggle to distinguish supposedly easy inputs from ID-data. We provide some illustrative unit-test results in Table 2.2 for a ViT pretrained on IN-21k and a ConvNext both with and without IN-21K pretraining. In general, most methods fail fewer unit tests when applied to pretrained models, however there are still many severely flawed combinations, often involving methods that would otherwise shine based on their detection of natural OOD data discussed above: especially the feature-based methods ViM, Maha and RMaha reveal weaknesses, each failing multiple unit-tests on at least 21 of 26 models. Many tested OOD detectors are vulnerable to *black*, *white* and *grey*, which is concerning as encountering inputs of this kind could occur in many real-world applications due to camera malfunction or occlusion. Here those feature-based methods only provide trustworthy results in combination with ViTs pretrained on IN-21k, the BiT-models and a pretrained EfficientNet-V2. Methods like Cos (7/26 models fail multiple tests) and MCM/RCos (7/26), originally designed for cosine-trained features as in CLIP, achieve remarkably strong OOD-detection performance on the unit-tests across a broad range of models, both with and without CLIP-pretraining. While taking note of these general trends, each OOD detector’s robustness to the OOD unit-tests should be examined individually.

Table 2.2: **Some detectors fail OOD unit-tests:** FPR for a ViT and a ConvNext (with and without pretraining) on selected unit-tests. FPR larger than 10% count as failed and are thus marked red. Especially for methods relying on feature representations (like ViM and Maha) the OOD unit-tests reveal difficulties.

	method	<i>bla</i>	<i>whi</i>	<i>gre</i>	<i>hor</i>	<i>SmN</i>	<i>Rad</i>	<i>mon</i>
ViT21k	MSP	0.0	0.0	0.0	0.2	0.5	0.0	0.0
	ViM	0.0	100.0	46.0	0.0	0.0	0.0	0.5
	Maha	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Cos	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Conv1k	MSP	0.0	0.0	0.0	60.5	0.8	0.0	0.0
	ViM	100.0	100.0	100.0	98.0	24.5	100.0	100.0
	Maha	100.0	100.0	100.0	87.5	27.5	100.0	100.0
	Cos	0.0	0.0	0.0	27.5	0.0	0.0	0.0
Conv21k	MSP	0.0	0.0	0.0	13.5	2.2	0.0	0.0
	ViM	100.0	100.0	100.0	0.0	0.0	41.2	0.5
	Maha	100.0	100.0	100.0	0.0	0.0	42.5	2.8
	Cos	0.0	0.0	0.0	0.0	0.0	0.0	0.0

2.6 Detailed results on NINCO

A detailed overview over the results on the NINCO benchmark is presented in Tables 2.3&2.4, where we show the mean FPR for all models and methods across the dataset’s OOD classes. The best method per model is marked bold, and the difference to the MSP-baseline is shown in green where a model outperforms the MSP-baseline and in red if it performs worse than MSP. It is clearly visible that there is no one-fits-all method. Instead, different models synergize with different methods. Overall, the two ViT models pretrained only on IN-21K in combination with Mahalanobis distance outperform other models and methods by a clear margin. This is in line with the observations of previous works (Koner et al., 2021; Fort et al., 2021; Galil et al., 2023), which also found the ViTs to perform exceptionally well. In terms of MSP, the ViTs are not better than e.g. the ConvNext, indicating that their improved OOD detection capabilities stem from a favourably structured feature-space. It is further interesting to see that for models without pretraining, out of all methods only Relative Mahalanobis and the cosine-based methods improve over the MSP-baseline fairly consistently. Apart from KL-Matching and KNN, most methods improve over the MSP-baseline for most pretrained models and the CLIP-methods Cosine and RCos perform comparably well, yielding their best results with models pretrained *both* on CLIP and IN-12k. Since CLIP models are trained with cosine-similarity, it is likely that the structure of the feature space after finetuning remains favorable to cosine-based methods, while it might harm the performance of other feature-based methods like Mahalanobis compared to models pretrained *only* on IN-21k.

2.7 Chapter conclusions

In this chapter, we introduced with NINCO a novel, ID-contamination-free and challenging OOD test-dataset for IN-1K with fine-grained class-resolution. We find that many OOD detectors work better than previously thought, when their recorded number of undetected OOD inputs is not inflated by ID contaminations. However, most detection methods cannot reliably be applied with arbitrary classifier models, as even OOD unit-tests are failed by many combinations. Also, even the strongest evaluated OOD detector still has a 27.5% FPR at 95% TPR, which arguably means that the OOD detection problem is only improved compared to MSP, but still far away from being solved at a satisfactory level.

Due to the issue that the OOD detection problem is not very well defined, however, it is difficult to assess what detection performance we can expect by future models. We are hopeful for NINCO and the cleaned test OOD subsets to facilitate the more precise development of reliable OOD detectors which do not try to avoid presumed failures which are actually correct decisions.

Recent developments

An updated Version 1.5 of the OpenOOD benchmark repository was released (Zhang et al., 2023), which removes some of the datasets for which we showed that they are not really OOD and includes NINCO without its class structure as a hard-OOD dataset in the benchmark. Yang et al. (2023) create the ImageNet-OOD dataset, a subset of IN-21 that uses the WordNet tree structure in order to avoid categorical ID contaminations.

Table 2.3: **Mean FPR on our NINCO dataset (1/2)**. Lower is better. The difference to MSP is shown in red if a method performs worse, and in green if it improves. Bold values mark the best-performing method per model.

pre	acc.	model	MSP	MaxL	Ener	KL-M	Maha	RMaha
21k	86.0	ViT-B-384	51.9	37.8 -14	36.9 -15	50.3 -2	27.5 -24	31.2 -21
	84.5	ViT-B-224	58.0	46.5 -12	46.1 -12	57.2 -1	31.9 -26	36.8 -21
	86.3	Swinv2-B-256	51.1	41.1 -10	40.0 -11	56.0 +5	62.8 +12	53.8 +3
	86.7	Deit3-B-384	61.8	56.0 -6	56.3 -5	60.3 -1	53.9 -8	48.8 -13
	85.7	Deit3-B-224	64.8	59.2 -6	58.1 -7	65.2 +0	60.0 -5	53.8 -11
	86.3	CnvNxt-B	47.2	41.1 -6	43.3 -4	54.9 +8	49.6 +2	42.4 -5
	84.1	CnvNxt-T	54.7	47.9 -7	45.4 -9	60.7 +6	46.9 -8	45.8 -9
	82.3	BiT-m	67.8	62.0 -6	63.2 -5	64.9 -3	50.0 -18	45.1 -23
	85.6	EffNetv2-M	50.7	48.3 -2	54.1 +3	54.6 +4	62.9 +12	51.6 +1
none	81.1	ViT-B-384	69.5	67.7 -2	68.1 -1	66.7 -3	60.0 -9	57.1 -12
	84.6	Swinv2-B-256	69.9	67.6 -2	72.2 +2	67.5 -2	63.9 -6	60.0 -10
	85.1	Deit3-B-384	67.3	72.8 +5	87.6 +20	64.6 -3	64.0 -3	59.4 -8
	83.8	Deit3-B-224	70.3	71.9 +2	82.3 +12	68.4 -2	69.0 -1	64.3 -6
	82.6	XCiT-M-224	72.7	73.3 +1	79.2 +6	71.8 -1	66.2 -6	63.5 -9
	84.3	XCiT-M-224-d	68.3	66.2 -2	73.1 +5	66.9 -1	66.4 -2	61.9 -6
	84.4	CnvNxt-B	64.7	71.5 +7	89.1 +24	68.0 +3	65.8 +1	60.6 -4
	78.0	BiT-s	78.8	81.2 +2	82.9 +4	68.4 -10	83.5 +5	64.1 -15
	85.1	EffNetv2-M	65.3	65.3 +0	74.5 +9	62.8 -2	62.5 -3	54.9 -10
	84.9	EffNetb7	66.8	69.0 +2	81.5 +15	62.7 -4	68.1 +1	54.6 -12
	77.7	EffNet-B0	72.0	72.4 +0	79.6 +8	72.3 +0	83.3 +11	74.0 +2
	80.4	ResNet50	72.4	74.3 +2	77.9 +6	69.0 -3	85.9 +13	69.5 -3
JFT	86.8	EffNetb7-ns	63.2	55.7 -7	61.5 -2	64.5 +1	87.4 +24	68.7 +6
clip +12k	87.2	ViT-B-384-l2b	50.2	47.4 -3	50.3 +0	52.2 +2	52.6 +2	47.3 -3
	87.0	ViT-B-384-oai	48.8	43.7 -5	44.1 -5	49.6 +1	57.7 +9	48.4 -0
clip	86.6	ViT-B-384-l2b	61.9	61.6 -0	65.8 +4	57.5 -4	52.7 -9	50.5 -11
	86.2	ViT-B-384-oai	64.9	64.9 +0	69.7 +5	61.8 -3	55.7 -9	53.7 -11
clip	74.3	clip-ViT-L-336	---	---	---	---	---	---
z. shot	66.6	clip-ViT-B-224	---	---	---	---	---	---

Table 2.4: **Mean FPR on our NINCO dataset (2/2)**. Lower is better. The difference to MSP is shown in red if a method performs worse, and in green if it improves. Bold values mark the best-performing method per model.

pre	acc.	model	MSP	ViM	E+R	KNN	Cos	MCM/RCos
21k	86.0	ViT-B-384	51.9	32.6 -19	38.5 -13	62.7 +11	46.0 -6	45.0 -7
	84.5	ViT-B-224	58.0	38.4 -20	49.4 -9	68.8 +11	54.7 -3	54.3 -4
	86.3	Swinv2-B-256	51.1	54.8 +4	37.4 -14	61.9 +11	51.4 +0	48.2 -3
	86.7	DeiT3-B-384	61.8	56.9 -5	51.6 -10	53.4 -8	48.4 -13	47.7 -14
	85.7	DeiT3-B-224	64.8	62.5 -2	55.2 -10	58.7 -6	54.2 -11	53.2 -12
	86.3	CnvNxt-B	47.2	41.5 -6	40.5 -7	51.8 +5	44.2 -3	42.6 -5
	84.1	CnvNxt-T	54.7	37.4 -17	44.1 -11	56.6 +2	51.2 -4	49.2 -5
	82.3	BiT-m	67.8	40.7 -27	57.1 -11	58.0 -10	51.6 -16	54.4 -13
	85.6	EffNetv2-M	50.7	53.5 +3	89.8 +39	67.5 +17	45.4 -5	50.6 -0
none	81.1	ViT-B-384	69.5	69.4 -0	65.8 -4	73.6 +4	68.7 -1	69.8 +0
	84.6	Swinv2-B-256	69.9	66.5 -3	68.8 -1	69.2 -1	63.5 -6	62.0 -8
	85.1	DeiT3-B-384	67.3	60.0 -7	90.2 +23	74.4 +7	67.1 -0	56.9 -10
	83.8	DeiT3-B-224	70.3	63.5 -7	83.1 +13	80.4 +10	73.0 +3	61.9 -8
	82.6	XCiT-M-224	72.7	64.9 -8	76.4 +4	71.8 -1	67.1 -6	66.0 -7
	84.3	XCiT-M-224-d	68.3	62.3 -6	72.4 +4	70.4 +2	64.6 -4	62.6 -6
	84.4	CnvNxt-B	64.7	65.4 +1	85.9 +21	70.5 +6	61.3 -3	58.6 -6
	78.0	BiT-s	78.8	73.5 -5	77.8 -1	83.2 +4	72.1 -7	84.1 +5
	85.1	EffNetv2-M	65.3	72.5 +7	69.6 +4	64.4 -1	59.6 -6	54.4 -11
	84.9	EffNetb7	66.8	72.7 +6	76.3 +10	66.8 +0	60.5 -6	53.7 -13
	77.7	EffNet-B0	72.0	75.2 +3	75.1 +3	86.9 +15	61.3 -11	69.8 -2
	80.4	ResNet50	72.4	78.6 +6	97.4 +25	77.9 +6	63.0 -9	62.1 -10
JFT	86.8	EffNetb7-ns	63.2	89.2 +26	61.7 -1	73.8 +11	65.2 +2	63.7 +1
clip +12k	87.2	ViT-B-384-l2b	50.2	45.8 -4	44.9 -5	45.4 -5	40.1 -10	40.2 -10
	87.0	ViT-B-384-oai	48.8	52.5 +4	42.2 -7	45.0 -4	39.3 -10	39.1 -10
clip	86.6	ViT-B-384-l2b	61.9	51.7 -10	63.2 +1	57.0 -5	50.8 -11	49.1 -13
	86.2	ViT-B-384-oai	64.9	56.9 -8	67.3 +2	61.4 -4	56.6 -8	54.3 -11
clip	74.3	clip-ViT-L-336	—	—	—	—	72.5	67.1
z. shot	66.6	clip-ViT-B-224	—	—	—	—	79.1	79.8

Part II

The Building Blocks of Out-of-Distribution Detectors

Chapter 3

Breaking Down Out-of-Distribution Detection

In the previous chapter we have defined the goal that OOD detectors aim to solve at test time. For this, we regarded the OOD detection problem on ImageNet where the scenario studied in the literature until this point has mostly studies post-hoc method, i.e. assuming the scenario that a model has been trained on as a standard classifier on the ID and a detection score is then derived from that model’s outputs and intermediate features. We now investigate how to obtain good OOD detectors when we can optimize them with respect to this task at train time, particularly if we have access to surrogate OOD data. In this and the following chapters, we regard in-distributions represented by smaller datasets like CIFAR-10, CIFAR-100 (both Krizhevsky et al. (2012)) and Restricted ImageNet (Tsipras et al., 2019), where besides post-hoc methods, the better feasibility of training full models has allowed the research community to consider approaches which make use of training techniques that incorporate surrogate OOD data.

This chapter is based on the paper “Breaking Down Out-of-Distribution Detection: Many Methods Based on OOD Training Data Estimate a Combination of the Same Core Quantities” (Bitterwolf et al., 2022), which we presented at ICML 2022 and in which we analyze the differences and similarities between different OOD detection methods. Alexander Meinke and I developed the idea of identifying and using the building blocks of OOD detection methods, formulated the definition of equivalence of OOD detection objectives and scores and proved several equivalences. I wrote most of the code for this project and conducted most experimental evaluations, with Alexander Meinke running those on Restricted ImageNet. Matthias Hein helped developing the project in in-depth discussion and with general guidance, and assisted with writing and revising. Maximilian Augustin helped writing the experimental results section. Simultaneously to this work on standard OOD detection for which I was the primary author, we used the underlying idea of functionally separating classifier and OOD detector for the adversarial OOD detection setting in the paper “Provably Robust Detection of Out-of-distribution Data (almost) for free” (Meinke et al., 2022), for which Alexander Meinke was the primary author and which we discuss in Chapter 6.

3.1 Goals for the analysis of OOD detectors

In the OOD detection setting, as we discussed before, at the time when models are trained and detection parameters are fitted there is by definition no information available on the distribution of OOD inputs that might appear once the model is used for inference. Nonetheless, it can often be reasonable to assume access to some additional data during training for which we know that it is from *some* out-distribution. We cannot, however, assume that this distribution is related to the OOD inputs that will be encountered at test-time. Still, such data can be useful in order to teach the model that there are some potential inputs that it cannot know about. When such data is available, we call it surrogate OOD training data.

One natural idea of using such surrogate OOD samples is enforcing low confidence of a classifier on such data, as it is done in Hendrycks et al. (2019a); Hein et al. (2019). However, there are other possible ways to leverage surrogate OOD data: A large number of different approaches to OOD detection based on combinations of density estimation, classifier confidence, classifier prediction entropy, logit space energy, feature space geometry, behaviour on auxiliary tasks, and other principles has been proposed. Many OOD detection papers are focused on establishing superior empirical detection performance and provide little theoretical background on differences but also similarities to existing methods. Here, we want to take a different path as we believe that a solid theoretical basis is needed to make further progress in this field. Our goal is to identify, at least for a particular subclass of techniques, whether the differences are indeed due to a different *underlying theoretical principle* or whether they are due to the efficiency of different *estimation techniques* for the same underlying detection criterion, called “scoring function”. In some cases, we will see that one can even disentangle the estimation procedure from the scoring function, so that one can simulate several different scoring functions from one model’s estimated quantities. A simple approach to OOD detection is to treat it as a binary discrimination problem between in- and out-of-distribution, or more generally predicting a score that monotonically corresponds with how likely the input is OOD. We show that from the perspective of Bayesian decision theory, several established methods are indeed equivalent to this binary discriminator. Differences arise mainly from i) the choice of the training out-distribution, e.g. the popular Outlier Exposure of Hendrycks et al. (2019a) has advocated the use of a rich and large set of natural images as a proxy for the distribution of natural images, and ii) differences in the estimation procedure. Concretely, the main contributions of the paper (Bitterwolf et al., 2022), which we discuss in this chapter, are:

- We show that several OOD detection approaches which optimize an objective that includes predictions on surrogate OOD data are equivalent to the binary discriminator between in- and out-distribution when analyzing the rankings induced by the Bayes optimal classifier/density.
- We derive the implicit scoring functions for the confidence loss (Lee et al., 2018a) used by Outlier Exposure (Hendrycks et al., 2019a), for Energy-Based OOD Detection (Liu et al., 2020), and for an extra background class for the out-distribution (Thulasidasan et al., 2021). The confidence scoring function turns out not to be equivalent to the “optimal” scoring

function of the binary discriminator when training and test out-distributions are the same.

- We show that the combination of a binary discriminator between in- and out-distribution with a standard classifier on the in-distribution, when trained in a shared fashion, yields OOD detection performance competitive with state-of-the-art methods based on surrogate OOD data.
- We show that density estimation is equivalent to discrimination between the in-distribution and uniform noise which indicates why standard density estimates are not suitable for OOD detection, as has frequently been observed.

Even though we identify that a simple baseline is competitive with the state-of-the-art, the main aim of this work is a better understanding of the key components of different OOD detection methods and to identify the key properties which lead to SOTA OOD detection performance. All of our findings are supported by extensive experiments on CIFAR-10 and CIFAR-100 with evaluation on various challenging out-of-distribution test datasets.

3.2 Models for OOD data and equivalence of OOD detection scores

We first characterize the set of transformations of a scoring function which leaves the OOD detection criteria like AUC or FPR invariant. This is important for the analysis later on, since the scoring functions of different methods are in many cases not identical as functions but yield the same OOD detection performance by those criteria. Like most work in the literature we consider OOD detection on a compact input domain X with the most important example being image classification where $X = [0, 1]^D$. The most popular approach to OOD detection is the construction of an in-distribution-scoring function $f : X \rightarrow \mathbb{R} \cup \{\pm\infty\}$ such that $f(x)$ tends to be smaller if x is drawn from an out-distribution $p(x|\mathcal{D}=o)$, short $p(x|o)$, than if it is drawn from the in-distribution $p(x|\mathcal{D}=i)$, short $p(x|i)$. There is a variety of different performance metrics for this task, with a very common one being the *area under the receiver-operator characteristic curve* (AUC). The AUC for a scoring function f distinguishing between an in-distribution $p(x|i)$ and an out-distribution $p(x|o)$ is given by

$$\begin{aligned} \text{AUC}_f(p(x|i), p(x|o)) \\ = \mathbb{E}_{\substack{x \sim p(x|i) \\ z \sim p(z|o)}} \left[\mathbb{1}_{f(x) > f(z)} + \frac{1}{2} \mathbb{1}_{f(x) = f(z)} \right]. \end{aligned}$$

We define an equivalence of scoring functions based on their AUCs and will show that this equivalence implies equality of other employed performance metrics as well.

Definition 1. Two scoring functions f and g are equivalent and we write $f \cong g$ if

$$\text{AUC}_f(p(x|i), p(x|o)) = \text{AUC}_g(p(x|i), p(x|o))$$

for all potential distributions $p(x|i)$ and $p(x|o)$.

As the AUC is not dependent on the actual values of f but just on the ranking induced by f one obtains the following characterization of the equivalence of two scoring functions.

Theorem 1. Two scoring functions f, g are equivalent $f \cong g$ if and only if there exists a strictly monotonously increasing $\phi : \text{range}(g) \rightarrow \text{range}(f)$ such that $f = \phi(g)$.

Proof. Assume that such a function ϕ exists. Then for any pair x, y we have the logical equivalences $g(x) > g(y) \Leftrightarrow f(x) = \phi(g(x)) > \phi(g(y)) = f(y)$ and $g(x) = g(y) \Leftrightarrow f(x) = \phi(g(x)) = \phi(g(y)) = f(y)$. This directly implies that the AUCs are the same, regardless of the distributions.

Assume $f \cong g$. For each $a \in \text{range}(g)$, choose some $\hat{a} \in g^{-1}(a)$. For any pair $x, y \in X$, by regarding the Dirac distributions $p(x|i) = \delta_x$ and $p(x|o) = \delta_y$ that are each concentrated on one of the points, we can infer that $f(x) > f(y) \Leftrightarrow \text{AUC}_f(p(x|i), p(x|o)) = 1 \Leftrightarrow \text{AUC}_g(p(x|i), p(x|o)) = 1 \Leftrightarrow g(x) > g(y)$ and similarly $f(x) = f(y) \Leftrightarrow g(x) = g(y)$. The latter ensures that the function defined as

$$\begin{aligned} \phi : \text{range}(g) &\rightarrow \text{range}(f) \\ a &\mapsto f(\hat{a}) \end{aligned} \tag{3.1}$$

is independent of the choice of $\hat{a}$ and that $f = \phi \circ g$, and the former confirms that ϕ is strictly monotonously increasing. $\square$

Corollary. The equivalence between scoring functions in Def. 1 is an equivalence relation.

Another metric is the *false positive rate at a fixed true positive rate q* , denoted as FPR@qTPR . A commonly used value for the TPR is 95%. The smaller the FPR@qTPR , the better the OOD discrimination performance.

Lemma 1. Two equivalent scoring functions $f \cong g$ have the same FPR@qTPR for any pair of in- and out-distributions $p(x|i), p(x|o)$ and for any chosen TPR q .

Proof. We know that a function ϕ as in Theorem 1 exists. Then for any pair x, y , we have the logical equivalences

$$g(x) > g(y) \Leftrightarrow f(x) = \phi(g(x)) > \phi(g(y)) = f(y) \tag{3.2}$$

and

$$g(x) = g(y) \Leftrightarrow f(x) = \phi(g(x)) = \phi(g(y)) = f(y) . \tag{3.3}$$

This directly implies that the FPR@qTPR-values are the same, for any $p(x|i), p(x|o)$ and q . $\square$

In the next section, we use the previous results to show that the Bayes optimal scoring functions of several proposed methods for out-of-distribution detection are equivalent to those of simple binary discriminators.

3.3 Bayes-optimal behaviour of binary discriminators and common OOD detection methods

In the following we will show that the Bayes optimal function of several existing approaches to OOD detection for unlabeled data are equivalent to a binary discriminator between in- and a (training) out-distribution, whereas different solutions arise for methods that involve labeled data. As the equivalences are based on the Bayes optimal solution, these are asymptotic statements and thus it has to be noted that convergence to the Bayes optimal solution can be infinitely slow and that the methods can have implicit inductive biases. This is why we additionally support our findings with extensive experiments.

3.3.1 OOD detection with methods using unlabeled data

We first provide a formal definition of OOD detection before we show the equivalence of density estimators resp. likelihood to a binary discriminator.

The OOD problem In order to make rigorous statements about the OOD detection problem we first have to provide the mathematical basis for doing so. We assume that we are given an in-distribution $p(x|i)$ and potentially also a *training* out-distribution $p(x|o)$. At this particular point no labeled data is involved, so both of them are just distributions over X . For simplicity we assume in the following that they both have a density wrt. the Lebesgue measure on $X = [0, 1]^d$. We assume that in practice we get samples from the mixture distribution

$$p(x) = p(x|i)p(i) + p(x|o)(1 - p(i))$$

where $p(i)$ is the probability that we expect to see in-distribution samples in total. In order to make the decision between in- and out-distribution for a given point x , it is thereby optimal to estimate

$$p(i|x) = \frac{p(x|i)p(i)}{p(x)} = \frac{p(x|i)p(i)}{p(x|i)p(i) + p(x|o)p(o)},$$

which is defined for all $x \in [0, 1]^d$ with $p(x) > 0$ (assuming $p(x|i)$ and $p(x|o)$ can be written as densities). If the training out-distribution is also the test out-distribution then this is already optimal but we would like that the approach generalizes to other unseen test out-distributions and thus an important choice is the training out-distribution $p(x|o)$. Note that as $p(i|x)$ is only well-defined for all x with $p(x) > 0$, it is thus reasonable to choose for $p(x|o)$ a distribution

with support in $[0, 1]^d$. In this case we ensure that the criterion with which we perform OOD detection is defined for any possible input x . This is desirable, as OOD detection should work for any possible input $x \in X$.

Optimal prediction of a binary discriminator between in- and out-distribution We consider a binary discriminator with model parameters θ between in- and (training) out-distribution, where $\hat{p}_\theta(i|x)$ is the predicted probability for the in-distribution. Under the assumption that $p(i)$ is the probability for in-distribution samples and using cross-entropy (which in this case is the logistic loss up to a constant global factor of $\log(2)$) the expected loss becomes:

$$\begin{aligned} \min_{\theta} \quad & p(i) \mathbb{E}_{x \sim p(x|i)} [-\log \hat{p}_\theta(i|x)] \\ & + p(o) \mathbb{E}_{x \sim p(x|o)} [-\log(1 - \hat{p}_\theta(i|x))] . \end{aligned}$$

One can derive that the Bayes optimal classifier minimizing the expected loss has the predictive distribution:

$$\hat{p}_{\theta^*}(i|x) = \frac{p(x|i)p(i)}{p(x|i)p(i) + p(x|o)p(o)} = p(i|x).$$

Thus at least for the training out-distribution, a binary classifier based on samples from in- and (training) out-distribution would suffice to solve the OOD detection problem perfectly.

Equivalence of density estimation and binary discrimination for OOD detection In this section we further analyze the relationship of common OOD detection approaches with the binary discriminator between in-and out-distribution. We start with density estimators sourced from generative models. A basic approach that is known to yield relatively weak OOD performance (Nalisnick et al., 2019; Ren et al., 2019; Xiao et al., 2020) is directly utilizing a model's estimate for the density $p(x|i)$ at a sample input x . An improved density based approach which uses perturbed in-distribution samples as a surrogate training out-distribution is the Likelihood Ratios method (Ren et al., 2019), which proposes to fit a generative model for both the in- and out-distribution and to use the ratio between the likelihoods output by the two models as a discriminative feature.

We show that with respect to the scoring function, the correct density $p(x|i)$ is equivalent to the Bayes optimal prediction of a binary discriminator between the in-distribution and uniform noise. Furthermore, the density ratio $\frac{p(x|i)}{p(x|o)}$ is equivalent to the prediction of a binary discriminator between the two distributions on which the respective models used for density estimation have been trained. Because of this equivalence, we argue that the use of binary discriminators is a simple alternative to these methods because of its easier training procedure.

We first prove the more general case of arbitrary likelihood ratios. In the following we use the abbreviation $\lambda = \frac{p(o)}{p(i)}$ to save space and make the statements more concise.

Lemma 2. Assume $p(x|i)$ and $p(x|o)$ can be represented by densities and the support of $p(x|o)$ covers the whole input domain X . Then $\frac{p(x|i)}{p(x|o)} \cong \frac{p(x|i)}{p(x|i) + \lambda p(x|o)}$ for any $\lambda > 0$.

Proof. The function $\phi : [0, \infty] \rightarrow [0, 1]$ defined by $\phi(x) = \frac{x}{x+\lambda}$ (setting $\phi(\infty) = 1$) fulfills the criterion from Theorem 1 of being strictly monotonously increasing. With

$$\phi\left(\frac{p(x|i)}{p(x|o)}\right) = \frac{\frac{p(x|i)}{p(x|o)}}{\frac{p(x|i)}{p(x|o)} + \lambda \frac{p(x|o)}{p(x|o)}} = \frac{p(x|i)}{p(x|i) + \lambda p(x|o)} \quad (3.4)$$

for $p(x|o) \neq 0$ and $\phi\left(\frac{p(x|i)}{0}\right) = \phi(\infty) = 1 = \frac{p(x|i)}{p(x|i) + \lambda \cdot 0}$, the equivalence follows. $\square$

This means that the likelihood ratio score of two optimal density estimators is equivalent to the in-distribution probability $\hat{p}_{\theta^*}(i|x)$ predicted by a binary discriminator and this is true for any possible ratio of $p(i)$ to $p(o)$. In the experiments below, we show that using such a discriminator has similar performance as the likelihood ratios of the different trained generative models.

For the approaches that try to directly use the likelihood of a generative model as a discriminative feature, this means that their objective is equivalent to training a binary discriminator against uniform noise, whose density is $p_{\text{Uniform}}(x) = p(x|o) = 1$ at any x .

Lemma 3. Assume that $p(x|i)$ can be represented by a density. Then $p(x|i) \cong \frac{p(x|i)}{p(x|i) + \lambda}$ for any $\lambda > 0$.

Proof. This is a special case of Lemma 2, by setting $p(x|o) = 1 = p_{\text{Uniform}}(x)$. $\square$

This provides additional evidence why a purely density based approach for many applications proves to be insufficient as an OOD detection score on the complex image domain: it is not reasonable to assume that a binary discriminator between certain classes of natural images on the one hand and uniform noise on the other hand provides much useful information about images from other classes or even about other nonsensical inputs.

3.3.2 OOD detection for methods using labeled data

We first discuss how one can formulate the OOD problem when one has access to labeled data for the in-distribution and we identify the target distribution of OOD detection using a background/reject class. Then we derive the Bayes optimal classifier of the confidence loss (Lee et al., 2018a) as used by the most successful variant of Outlier Exposure (Hendrycks et al., 2019a) and discuss the implicit scoring function. In most cases the scoring functions turn out not to be equivalent to $p(i|x)$ (which is optimal if training and test out-distribution agree) as they integrate additional information from the classification task. Given a joint in-distribution $p(y, x|i)$ (where $y \in \{1, \dots, K\}$ given that we have K labels) for the labeled in-distribution, there are different ways how to come up with a joint distribution for in- and out-distribution. Interestingly, the different encodings used e.g. in training with a background class (Thulasidasan et al., 2021) vs. training a classifier with confidence loss (Lee et al., 2018a)

together with variants of the employed scoring function lead to methods which unexpectedly can have quite different behavior.

Background class: In this case we just put all out-of-distribution samples into a $K + 1$ -class which is typically called background/reject class (Thulasidasan et al., 2021). The joint distribution then becomes

$$p(y, x) = \begin{cases} p(y, x|i)p(i) & \text{if } y \in \{1, \dots, K\}, \\ p(x|o)p(o) & \text{if } y = K+1. \end{cases}$$

We denote by $p(x|i) = \sum_{y=1}^K p(y, x|i)$ the marginal in-distribution and note that the marginal distribution of the joint distribution of in- and out-distribution is again

$$p(x) = p(x|i)p(i) + p(x|o)p(o).$$

Thus we get the conditional distribution

$$p(y|x) = \begin{cases} p(y|x, i)p(i|x) & \text{if } y \in \{1, \dots, K\}, \\ p(o|x) = 1 - p(i|x) & \text{if } y = K+1. \end{cases}$$

The Bayes optimal solution of training with a background class using any calibrated loss function $L(y, f(x))$, e.g. the cross-entropy loss (Laptev et al., 2016), then yields a Bayes optimal classifier f^* which has a predictive distribution $p_{f^*}(y|x) = p(y|x)$. There are two potential scoring function that come to mind:

$$s_1(x) = 1 - p_{f^*}(K+1|x) \text{ and } s_2(x) = \max_{k=1, \dots, K} p_{f^*}(k|x)$$

The first one, used in Chen et al. (2021); Thulasidasan et al. (2021), is motivated by the fact that $p_{f^*}(K+1|x)$ is directly the predicted probability that the point is from the out-distribution as indeed it holds: $s_1(x) = p(i|x)$ which is the optimal scoring function if training and test out-distribution are equal. On the other hand the maximal predicted probability $\max_{k=1, \dots, K} p_{f^*}(k|x)$, which is often employed as a scoring function (Hendrycks & Gimpel, 2017), becomes for the Bayes optimal classifier

$$s_2(x) = p(i|x) \max_{k=1, \dots, K} p(k|x, i),$$

which is a product of $p(i|x)$ and the maximal conditional probability of some class of the in-distribution; note that s_2 is well defined as $p(i|x)$ is defined if $p(x|o)$ has support everywhere in X and if $p(i|x) > 0$ then also $p(x|i) > 0$. Thus the scoring function $s_2(x)$ integrates additionally to $p(i|x)$ also class-specific information and is thus less dependent on the chosen training out-distribution. In fact, one can see that s_2 only ranks points high if both the binary discriminator *and* the classifier rank the corresponding point high. However, in the case where training and test out-distribution are identical, this scoring function is not equivalent to $p(i|x)$ and thus

introduces a bias in the estimation.

Outlier Exposure (Hendrycks et al., 2019a) with confidence loss (Lee et al., 2018a): We analyze the Bayes optimal solution for the confidence loss (Lee et al., 2018a) that is used by Outlier Exposure (OE) and show that the associated scoring function can be written, similarly to the scoring function $s_2(x)$ for training with a background class, as a function of $p(i|x)$ and $p(y|x, i)$.

The training objective with the confidence loss is in expectation given by

$$\min_{\theta} \mathbb{E}_{\substack{(x,y) \sim \\ p(x,y|i)}} [\mathcal{L}_{\text{CE}}(f_{\theta}(x), y)] + \lambda \mathbb{E}_{x \sim p(x|o)} [\mathcal{L}_{\text{CE}}(f_{\theta}(x), u^K)] ,$$

where θ are the model parameters and $f_{\theta}(x) \in \mathbb{R}^K$ is the model output as logits, and $u^K = (\frac{1}{K}, \dots, \frac{1}{K})^T$ is the uniform distribution over the K classes of the in-distribution classification task.

In the following theorem we derive the Bayes optimal predictive distribution for this training objective.

Theorem 2. *The predictive distribution $p_{f^*}(y|x)$ of the Bayes optimal classifier f^* minimizing the expected confidence loss is given for $y \in \{1, \dots, K\}$ as*

$$p_{f^*}(y|x) = p(i|x)p(y|x, i) + \frac{1}{K}(1 - p(i|x)) .$$

Proof. Minimizing the loss of Outlier Exposure

$$\min_{\theta} \mathbb{E}_{\substack{(x,y) \sim \\ p(x,y|i)}} [\mathcal{L}_{\text{CE}}(f_{\theta}(x), y)] + \lambda \mathbb{E}_{x \sim p(x|o)} [\mathcal{L}_{\text{CE}}(f_{\theta}(x), u^K)] , \quad (3.5)$$

means solving the optimization problem

$$\begin{aligned} \min_{p_f(\cdot|x)} & -p(i|x) \cdot \sum_{k=1}^K p(k|x, i) \cdot \log p_f(k|x) - (1 - p(i|x)) \cdot \sum_{k=1}^K \frac{1}{K} \cdot \log p_f(k|x) \\ \text{subject to} & \quad p_f(k|x) \geq 0 \text{ for each } k \in \{1, \dots, K\} \\ & \quad \sum_{k=1}^K p_f(k|x) = 1 , \end{aligned} \quad (3.6)$$

where $p_f(\cdot|x)$ is the model's K -dimensional prediction. For $p(i|x) = 0$ or $p(k|x, i) = 0$, the optimalities of the respective terms are easy to show (applying the common conventions for $0 \log 0$), so we assume that those are non-zero. The Lagrange function of the optimization

problem is

$$\begin{aligned}
 L(p_f(\cdot|x), \alpha, \beta) = & -p(i|x) \cdot \sum_{k=1}^K p(k|x, i) \cdot \log p_f(k|x) - (1 - p(i|x)) \cdot \sum_{k=1}^K \frac{1}{K} \cdot \log p_f(k|x) \\
 & - \sum_{k=1}^K \alpha_k p_f(k|x) + \beta \left(-1 + \sum_{k=1}^K p_f(k|x) \right), \tag{3.7}
 \end{aligned}$$

with $\beta \in \mathbb{R}$ and $\alpha \in \mathbb{R}_+^K$. Its first derivative with respect to $p_f(k|x)$ for any k is

$$\begin{aligned}
 \frac{\partial L}{\partial p_f(k|x)} = & -p(i|x) \cdot p(k|x, i) \frac{1}{p_f(k|x)} - (1 - p(i|x)) \cdot \frac{1}{K} \frac{1}{p_f(k|x)} - \alpha_k + \beta \\
 = & -\frac{s^K(k|x)}{p_f(k|x)} - \alpha_k + \beta, \tag{3.8}
 \end{aligned}$$

where we set $s^K(k|x) := p(i|x)p(k|x, i) + \frac{1}{K}(1 - p(i|x))$. The second derivative is a positive diagonal matrix for any point of its domain, therefore we find the unique minimum by setting (3.8) to zero, i.e. at

$$p_f(k|x) = \frac{s^K(k|x)}{\beta - \alpha_k}. \tag{3.9}$$

The dual problem is hence the maximization (with $\alpha_k \geq 0$) of

$$\begin{aligned}
 q(\alpha, \beta) = & -p(i|x) \cdot \sum_{k=1}^K p(k|x, i) \cdot \log \frac{s^K(k|x)}{\beta - \alpha_k} - (1 - p(i|x)) \cdot \sum_{k=1}^K \frac{1}{K} \cdot \log \frac{s^K(k|x)}{\beta - \alpha_k} \\
 & - \sum_{k=1}^K \alpha_k \frac{s^K(k|x)}{\beta - \alpha_k} + \beta \left(-1 + \sum_{k=1}^K \frac{s^K(k|x)}{\beta - \alpha_k} \right) \\
 = & \sum_{k=1}^K s^K(k|x) \left(-\log s^K(k|x) + \log(\beta - \alpha_k) + \frac{\beta}{\beta - \alpha_k} - \frac{\alpha_k}{\beta - \alpha_k} \right) - \beta;
 \end{aligned}$$

here, α only appears in $\log(\beta - \alpha_k)$ which has a positive factor $s^K(k|x)$, so $\alpha = 0$ maximizes the expression. Noting $\sum_{k=1}^K s^K(k|x) = 1$, what remains is $q^0(\beta) = 1 + \log(\beta) - \sum_{k=1}^K s^K(k|x) \log s^K(k|x) - \beta$, which is maximized by $\beta = 1$. This means that the dual optimal pair is $p_f(k|x) = s^K(k|x)$, $(\beta = 1, \alpha = 0)$. Slater's condition (Boyd & Vandenberghe, 2004) holds since the feasible set of the original problem is the probability simplex. Thus, $p_{f*}(\cdot|x) = s^K(x)$ is indeed primal optimal. $\square$

Thus the effective scoring function of using the probability of the predicted class as sug-

gested in Hendrycks & Gimpel (2017); Lee et al. (2018a); Hendrycks et al. (2019a) is

$$\begin{aligned} s_3(x) &= p(i|x) \max_{y=1,\dots,K} p(y|x, i) + \frac{1}{K} (1 - p(i|x)) \\ &= p(i|x) \left[\max_{y=1,\dots,K} p(y|x, i) - \frac{1}{K} \right] + \frac{1}{K}. \end{aligned}$$

Please note that the term inside the brackets is positive as $\max_{k=1,\dots,K} p(k|x, i) \geq \frac{1}{K}$. Interestingly, the scoring functions s_2 and s_3 are not equivalent even though they look quite similar. In particular, due to the subtraction of $\frac{1}{K}$ the scoring function s_3 puts more emphasis on the classifier than s_2 .

Energy-Based OOD Detection (Liu et al., 2020): With Energy-Based OOD Detection (Liu et al., 2020), we exhibit a surprising case of the equivalence of Binary Discriminators to a further OOD detection method which is based on ideas quite different to those of the other extensively discussed methods. This method uses the energy of a model’s logit prediction which is defined as $E_\theta(x) = -\log \sum_{y=1}^K e^{f_\theta(y|x)}$ as an OOD detection score. The model’s energy is fine-tuned to be low on the in-distribution and high on surrogate OOD data, within certain margins $m_{\text{out}}, m_{\text{in}}$. Note that this optimization is only done in the scenario where surrogate OOD data is available. In the previous chapter where we evaluated it on NINCO, we regard the post-hoc method which assumes that models with standard training without OOD data already has a (weaker) tendency to display lower energy on ID data.

Energy-Based OOD Detection is based on the premise that the logits $f_\theta(x)$ of a classifier can be used to define an energy (here we ignore a potential temperature factor which as Liu et al. (2020) find can in good conscience be set to one)

$$E_\theta(x) = -\log \sum_{l=1}^K e^{f_\theta(l|x)} \quad (3.10)$$

which the model assigns to the input, as has been proposed by LeCun et al. (2006); Grathwohl et al. (2020). Ideally, this energy would be equivalent to the probability density of the in-distribution via

$$\log p(x|i) = -E_\theta(x) - Z \quad \text{where } Z = \int_{z \in [0,1]^D} e^{-E_\theta(z)} dz. \quad (3.11)$$

However, since the integral over the whole image domain is intractible, it is not possible to effectively decrease $-E_\theta$ on the in-distribution directly while also controlling Z . Naïvely training the classifier to have high energy on random inputs, i.e. uniform noise, is of course not a solution, since the model easily distinguishes the noise, and it is very unlikely to encounter any at least vaguely real images within finite training time. Thus, Liu et al. (2020) rather use a surrogate training out-distribution of natural images which they increase the energy on during training; for their experiments, they take 80 Million Tiny Images. Simultaneously, they minimize the standard classifier cross-entropy on the in-distribution. In order to avoid

infinitely small potential losses, their training objective uses two margin hyper-parameters $m_{\text{in}} < m_{\text{out}}$ and reads

$$\begin{aligned} & \mathbb{E}_{(x,y) \sim p(x,y|i)} -\log p_{\theta}(y|x) \\ & + \lambda \cdot \left(\mathbb{E}_{(x) \sim p(x|i)} \max \{0, E_{\theta}(x) - m_{\text{in}}\}^2 + \mathbb{E}_{w \sim p(w|o)} \max \{0, m_{\text{out}} - E_{\theta}(w)\}^2 \right) \end{aligned} \quad (3.12)$$

with

$$\log p_{\theta}(y|x) = \log \frac{e^{f_{\theta}(y|x)}}{\sum_{l=1}^K e^{f_{\theta}(l|x)}} = f_{\theta}(y|x) + E_{\theta}(x) . \quad (3.13)$$

Note that their λ does not balance between in and out by depending on a prior on $p(i)$, but it balances the energy regularization compared to the classifier cross-entropy loss.

Theorem 3. *The Bayes optimal logit output $f_{\theta}^*(x)$ of the Energy-Based OOD detection model minimizing the expected loss on an input x yields the same class probabilities $p_{\theta}^*(k|x) = p(k|x, i)$ that are optimal for a standard classifier with cross-entropy loss and simultaneously fulfills*

$$-E_{\theta}^*(x) = p(i|x) \cdot (m_{\text{out}} - m_{\text{in}}) - m_{\text{out}} .$$

Proof. Our goal is to find for a given input x the model output $f_{\theta}(x)$ that minimizes the expected loss, assuming that we know the probabilities $p(y|x, i)$ and $p(i|x)$. This expected loss is

$$-p(i|x) \cdot \sum_{k=1}^K p(k|x, i) \log p_{\theta}(k|x) \quad (3.14)$$

$$+ \lambda \cdot \left(p(i|x) \max \{0, E_{\theta}(x) - m_{\text{in}}\}^2 + (1 - p(i|x)) \max \{0, m_{\text{out}} - E_{\theta}(x)\}^2 \right) . \quad (3.15)$$

First, we note that if a certain $E_{\theta}^*(x)$ minimizes the expected energy loss to

$$\lambda \cdot \left(p(i|x) \max \{0, E_{\theta}^*(x) - m_{\text{in}}\}^2 + (1 - p(i|x)) \max \{0, m_{\text{out}} - E_{\theta}^*(x)\}^2 \right) , \quad (3.16)$$

and if some $f_{\theta}^{\sharp}(x)$ minimizes the expected CE loss (note that its minimization is independent of the positive factor $p(i|x)$) to

$$-\sum_{k=1}^K p(k|x, i) \log \frac{e^{f_{\theta}^{\sharp}(k|x)}}{\sum_{l=1}^K e^{f_{\theta}^{\sharp}(l|x)}} , \quad (3.17)$$

with corresponding $E_{\theta}^{\sharp}(x) = -\log \sum_{i=1}^K e^{f_{\theta}^{\sharp}(x)[i]}$, then the logit output with $-E_{\theta}^*(x) + E_{\theta}^{\sharp}(x)$

added to each component has the energy

$$E_{\theta}(x) = -\log \sum_{i=1}^K e^{f_{\theta}^{\sharp}(x)[i] - E_{\theta}^*(x) + E_{\theta}^{\sharp}(x)} = -\log \left(e^{-E_{\theta}^*(x) + E_{\theta}^{\sharp}(x)} \cdot \sum_{i=1}^K e^{f_{\theta}^{\sharp}(x)[i]} \right) \quad (3.18)$$

$$= E_{\theta}^*(x) - E_{\theta}^{\sharp}(x) - \log \sum_{i=1}^K e^{f_{\theta}^{\sharp}(x)[i]} = E_{\theta}^*(x), \quad (3.19)$$

which means that this $f^*(x)$ minimizes the expected energy loss, and also fulfills

$$p_{\theta}(x)[y] = \frac{e^{f_{\theta}^{\sharp}(x)[y] - E_{\theta}^*(x) + E_{\theta}^{\sharp}(x)}}{\sum_{l=1}^K e^{f_{\theta}^{\sharp}(x)[l] - E_{\theta}^*(x) + E_{\theta}^{\sharp}(x)}} = \frac{e^{f_{\theta}^{\sharp}(x)[y]}}{\sum_{l=1}^K e^{f_{\theta}^{\sharp}(x)[l]}}, \quad (3.20)$$

i.e. has the same probability predictions as $f^{\sharp}$ and thus also minimizes the expected CE loss. This means that both can be optimized independently.

As we've seen before, the optimal $p_{\theta}(k|x)$ for the expected CE loss is $p_{\theta}(k|x) = p(k|x, i)$.

Thus we independently minimize the expected energy loss

$$p(i|x) \max \{0, E_{\theta}(x) - m_{\text{in}}\}^2 + (1 - p(i|x)) \max \{0, m_{\text{out}} - E_{\theta}(x)\}^2. \quad (3.21)$$

The first derivative with respect to $E_{\theta}(x)$ is

$$\begin{cases} 2 \cdot (1 - p(i|x)) \cdot (E_{\theta}(x) - m_{\text{out}}) & \text{for } E_{\theta}(x) < m_{\text{in}}, \\ 2 \cdot (p(i|x) \cdot (E_{\theta}(x) - m_{\text{in}}) + (1 - p(i|x)) \cdot (E_{\theta}(x) - m_{\text{out}})) & \text{for } m_{\text{in}} < E_{\theta}(x) < m_{\text{out}}, \\ 2 \cdot p(i|x) \cdot (E_{\theta}(x) - m_{\text{in}}) & \text{for } m_{\text{out}} < E_{\theta}(x). \end{cases} \quad (3.22)$$

which simplified is

$$\begin{cases} < 0 & \text{for } E_{\theta}(x) < m_{\text{in}}, \\ 2 \cdot (E_{\theta}(x) + p(i|x) \cdot (m_{\text{out}} - m_{\text{in}}) - m_{\text{out}}) & \text{for } m_{\text{in}} < E_{\theta}(x) < m_{\text{out}}, \\ > 0 & \text{for } m_{\text{out}} < E_{\theta}(x). \end{cases} \quad (3.23)$$

Here, for simplicity we again make the reasonable assumption that in- and out-distribution have full support, i.e. $0 < p(i|x) < 1$. (If we do not want to make the full support assumptions, for $p(i|x) = 0$, any energy $\geq m_{\text{out}}$ would be optimal, and for $p(i|x) = 1$, any energy $\leq m_{\text{in}}$ would be optimal.)

At the margin $E_{\theta}(x) = m_{\text{in}}$, the derivatives for $E_{\theta}(x) < m_{\text{in}}$ and $m_{\text{in}} < E_{\theta}(x) < m_{\text{out}}$ coincide as $2 \cdot (1 - p(i|x)) \cdot (m_{\text{in}} - m_{\text{out}}) < 0$, which means that the minimizer is $> m_{\text{in}}$. Similarly, we see that the minimizer is $< m_{\text{out}}$.

As the second derivative is positive, by solving where the derivative is zero, we find that the

optimal negative energy (which is the score which they use for OOD detection) is

$$-E_{\theta}^*(x) = p(i|x) \cdot (m_{\text{out}} - m_{\text{in}}) - m_{\text{out}} . \quad (3.24)$$

□

This is a strictly monotonously increasing function in $p(i|x)$. By our Theorem 1, we conclude that the negative energy score they obtain from their method is (for the Bayes optimal model that minimizes the loss on the training distributions) equivalent to $p(i|x)$.

Corollary. *The Bayes optimal solution of the Energy-based OOD detection criterion is equivalent to the Bayes optimal solution of a binary discriminator between the training in- and out-distributions.*

Note that while the classifier and energy loss terms *can* be optimized independently, the model is trained for both tasks simultaneously, which means that similar synergies to those we observe in 3.2 might contribute to the good performance of this method. While we observe slightly worse results when transferring the method to using AutoAugment and OpenImages for fine-tuning in 3.6, the empirical evaluation of Energy-Based models overall shows similar behaviour to the binary discriminator between in- and training-out-distribution in practice.

3.3.3 Separate vs. shared estimation of $p(i|x)$ and $p(y|x, i)$

So far we have derived that at least from the point of view of the ranking induced by the Bayes optimal solution, OOD detection based on generative methods, likelihood ratios, logit energy, and the background class formulation with the scoring function s_1 is equivalent to a binary classification problem between in- and out-distribution in order to estimate $p(i|x)$. The differences arise mainly in the choice of the training out-distribution $p(x|o)$: i) uniform for generative resp. density based methods, ii) a quite specific out-distribution for likelihood ratios (Ren et al., 2019) and iii) a proxy of the distribution of all natural images (Hendrycks et al., 2019a; Thulasidasan et al., 2021). On the other hand when labeled data is involved we can additionally train a classifier on the in-distribution in order to estimate $p(y|x, i)$. We will then combine the estimates of $p(i|x)$ and $p(y|x, i)$ according to the three scoring functions derived in the previous section and check if the novel OOD detection methods constructed in this way perform similar to the OOD methods from which we derived the corresponding scoring function i) OOD detection with a background class (Thulasidasan et al., 2021) or ii) using Outlier Exposure (Hendrycks et al., 2019a). This will allow us to differentiate between differences of the employed scoring functions for OOD detection and the estimators for the involved quantities. In this way we foster a more systematic approach to OOD detection.

In the unlabeled case we train simply the binary classifier $p_{\theta} : [0, 1]^d \rightarrow \mathbb{R}$ using logis-

tic/cross entropy loss in a class balanced fashion

$$\min_{\theta} \left(-\frac{1}{N} \sum_{i=1}^N \log(\hat{p}_{\theta}(i|x_i^{\text{IN}})) - \frac{\lambda}{M} \sum_{j=1}^M \log(1 - \hat{p}_{\theta}(i|x_j^{\text{OUT}})) \right),$$

where $(x_i^{\text{IN}})_{i=1}^N$ and $(x_j^{\text{OUT}})_{j=1}^M$ are samples from the in-distribution and the out-distribution.

In the case where we have labeled data we can additionally solve the classification problem. The obvious approach is to train the binary classifier for estimating $p(i|x)$ and the classifier to estimate $p(y|x, i)$ completely independently. Not surprisingly, we show in Section 3.6 that this approach works less well. In fact both tasks benefit from each other. Moreover, in training a neural network using a background class or with Outlier Exposure (Hendrycks et al., 2019a) we are implicitly using a shared representation for both tasks which improves the results.

Thus we propose to train the binary discriminator of in-versus out-distribution together with the classifier on the in-distribution jointly. Concretely, we use a neural network with $K + 1$ outputs where the first K outputs represent the classifier and the last output is the logit of the binary discriminator. The resulting shared problem can then be written as

$$\min_{\theta} \left(-\frac{1}{N_b} \sum_{r=1}^{N_b} \log \hat{p}_{\theta}(i|x_r^{\text{IN}}) - \frac{\lambda}{M} \sum_{s=1}^M \log(1 - \hat{p}_{\theta}(i|x_s^{\text{OUT}})) - \frac{1}{N_c} \sum_{t=1}^{N_c} \log \hat{p}_{\theta}(y_t^{\text{IN}}|x_t^{\text{IN}}) \right)$$

where $\lambda = \frac{p(o)}{p(i)}$ which is typically set to 1 during training in order to get a class-balanced problem. Note that the in-distribution samples $(x_r^{\text{IN}})_{r=1}^{N_b}$ used to estimate $p(i|x)$ can be a superset of the labeled examples $(x_t^{\text{IN}}, y_t^{\text{IN}})_{t=1}^{N_c}$ used to train the classifier so that one can potentially integrate unlabeled data - this is an advantage compared to OOD detection with a background class or Outlier Exposure where this is not directly possible. We stress that the loss functions of the classifier and the discriminator act on independent outputs; the functions modelling the two tasks only interact with each other due to the shared network weights up to the final layer. Nevertheless, we see in the next Section 3.6 that training with a shared representation boosts both the classifier and the binary discriminator.

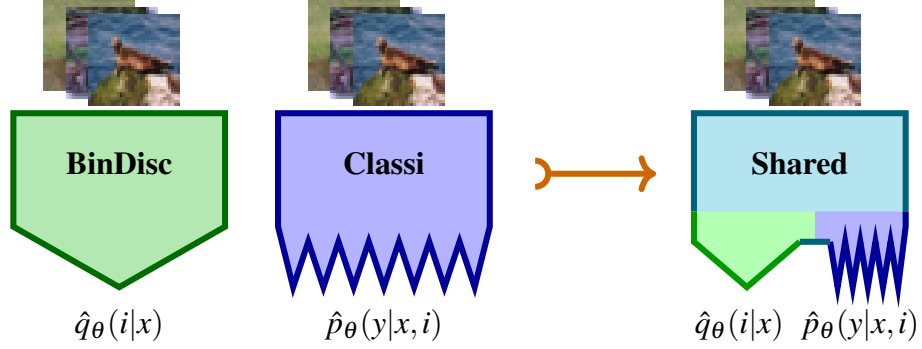


Figure 3.1: The classifier estimating $p(y|x, i)$ and the binary discrimination between in- and training out-distribution estimating $p(i|x)$ can be treated by **individual models**. Their performance can be greatly enhanced via **shared training**, since having one head each allows for a shared feature representation.

3.4 Confidence loss models and models with a background class or a binary discriminator are only equivalent after s_3 is applied

Seeing that for both in-distribution accuracy and OOD detection, OE models trained with confidence loss, models with background class and shared classifier/discriminator combinations behave very similarly, the question arises if the training methods themselves lead to equivalent models. One idea might be that an effect of the confidence loss on the degree of freedom from logit translation invariance could be unfolded to obtain a $K + 1$ -dimensional output that contains the same information as a classifier with background class or with an additional binary discriminator output (the latter two are indeed equivalent). This is not the case, as the following example that in certain situations, the s_1 and s_2 scores of background class models and classifier/discriminator combinations are able to separate in- and out-distribution, while the s_3 score and the equivalent confidence loss/OE models cannot.

3.4.1 Example problem: Euro coin classifier

As an example where the mentioned non-equivalence would occur, we hypothetically regard the task of classifying photos of 1-Euro coins by the issuing country. Each €1 coin features a common side that is the same for each country and a national side that pictures a unique motive per country. We assume that one side is visible on each photo, and that the training dataset of size $2mK$ is balanced, consisting of $2m$ coin photos with label c for each country $c \in \{1, \dots, K\}$, where m photos show the common side and the other m photos show the informative national side for each country c .

It is easy to see that the Bayes optimal classifier trained with cross-entropy loss on this

dataset predicts 100% for the respective country c when shown a photo of the national side of a €1 coin, and predicts $1/K$ for each country when shown the common side of a €1 coin.

Now we compare the behaviour of the different methods given a training out-distribution of poker chips images which are clearly recognizable as not being €1 coins.

A K -class model trained with confidence loss (Lee et al., 2018a) will not make a difference between common side coin images and poker chip images, and in the Bayes optimal case, it will predict the uniform class distribution in both cases. This does not only hold for the prediction of a hypothetical Bayes optimal model: assuming full batch gradient descent and identical sets of m common side training photos for each class, the loss for a common side input is the same as the loss for a poker chip.

On the other hand, a binary discriminator will easily distinguish between poker chips and €1 coins, no matter which side of the coin is shown. The same holds for a model with background class: the score of the class $K + 1$ will be close to 1 for chips and close to 0 for €1 coins.

We conclude that in the described situation, models trained with confidence loss/outlier exposure are not able to sufficiently distinguish in- and out-distribution, while the s_1 scoring function of a classifier with background class or a binary discriminator is suitable for this task.

With the s_2 scoring function, the background class model gives us $s_2(x) = \max_{k=1,\dots,K} p_f(k|x)$, and thus $\frac{(1-p_f(K+1|x))}{K} \leq s_2(x) \leq 1 - p_f(K+1|x)$, which means that if $p_f(K+1|x)$ is sufficiently large for in-distribution inputs and sufficiently small for out-of-distribution inputs, s_2 is able to distinguish them independent of inconclusiveness in the first K classes. Similarly, s_2 applied to a binary discriminator with a classifier (shared trained or not) will be able to distinguish common sides of coins and poker chips.

With s_3 , on the other hand, common sides of coins and poker chips can no longer be separated. For a classifier/discriminator pair, as defined above, $s_3(x) = p_d(i|x) \left[\max_{y=1,\dots,K} p_c(y|x, i) - \frac{1}{K} \right] + \frac{1}{K}$. If on the common side of a coin the classifier predicts uniform $\frac{1}{K}$, we have $s_3(x) = \frac{1}{K}$ no matter what the discriminator $p_d(i|x)$ predicts. On poker chips with discriminator prediction $p_d(i|x) = 1$, we also get $s_3(x) = 1$. For background class models, $s_3(x) = \max_{k=1,\dots,K} p_f(k|x) + \frac{1}{K} p_f(K+1|x)$ also yields $\frac{1}{K}$ for a common side where the prediction over the K in-distribution classes is uniform and for a poker chips, where $p_f(K+1|x) = 1$. The fact that in this coin scenario when scored with s_3 , background class and classifier/discriminator combinations have the same problem as confidence loss/OE is not surprising considering their equivalence shown in Theorem 2.

3.5 An overview of CIFAR scale OOD detection methods

Out-of-distribution detection has been an important research area in recent years, and several approaches that are fitted towards different training and inference scenarios have been

proposed.

One seemingly obvious line of thought is to use generative models for density estimation to differentiate between in- and out-distribution (Bishop, 1994; Nalisnick et al., 2019; Ren et al., 2019; Nalisnick et al., 2019; Xiao et al., 2020). Recent methods to a certain extent overcome the problem mentioned in Nalisnick et al. (2019) that generative models can assign higher likelihood to distributions on which they have not been trained. Another line of work are score-based methods using an underlying classifier or the internal features of such a classifier, potentially combined with a generative model (Hendrycks & Gimpel, 2017; Liang et al., 2018; Lee et al., 2018b; Hendrycks et al., 2019a; Hein et al., 2019). One of the most effective methods up to now is Outlier Exposure (Hendrycks et al., 2019a) and work building upon it (Chen et al., 2021; Meinke & Hein, 2020; Mohseni et al., 2020; Augustin et al., 2020; Papadopoulos et al., 2021; Thulasidasan et al., 2021) where a classifier is trained on the in-distribution task and one enforces low confidence as proposed by Lee et al. (2018a) during training on a large and diverse set of out-of-distribution images (Hendrycks et al., 2019a) which can be seen as a proxy of all natural images. This approach generalizes well to other out-distributions. Recently, NTOM (Chen et al., 2021) has achieved excellent results for detecting far out-of-distribution data by adding a background class to the classifier which is trained on samples from the surrogate out-distribution that are mined such that they show a desired hardness for the model. At test time, the output probability for that class is used to decide if an input is to be flagged as OOD. Their ATOM method does the same while also adding adversarial perturbations to the OOD inputs during training. Even though it has been claimed that new approaches outperform Hendrycks et al. (2019a), up to our knowledge this has not been shown consistently across different and challenging test out-of-distribution datasets (including close and far out-of-distribution datasets). Below, we discuss some other recently proposed approaches that build upon different premises on the data available during training.

Hendrycks et al. (2019b) do not use any OOD data during training and instead teach the model to predict whether an input has been rotated by 0° , 90° , 180° or 270° . For inference, they use the loss of this predictor as an OOD score, and add this score to classifier output entropy, which behaves very similar to classifier confidence. Similar to our methods, they also use shared representations and the combination of the in-distribution classifier with a dedicated OOD detection score. If one interprets their rotation predictor loss as being an estimator of $\log p(o|x)$ for some implicit out-distribution, their scoring function coincides with our s_2 scores.

Golan & El-Yaniv (2018) learn a similar transformation detector (with Dirichlet statistics collected on the in-distribution replacing ground truth labels) and use it directly to detect OOD samples without using in-distribution class information.

Winkens et al. (2020) fit for each class a normalized Mahalanobis detector on the activations of a model trained with SimCLR and a classification head on only the in-distribution with smoothed labels. They describe their method as applying class-wise density estimation in the feature space, where the normalized Mahalanobis distance is equivalent to a Gaussian density for each class.

Roy et al. (2021) treat an interesting application of flagging unseen skin diseases, making

use of class labels that are also available for their training OOD data, which contains diseases that are different from both the in-distribution diseases and the unseen diseases. This allows them to do fine-grained OOD detection by regarding the sum over all OOD classes which for their dataset shows large improvement over methods that treat the training out-distribution as one class. They gain additional slight improvements by combining this with a coarse grained binary loss that treats the sum over all in-distribution class probabilities as $p(i|x)$ and the sum over all OOD classes as $p(o|x)$. They show that this method can be combined with various representation learning approaches in order to improve their detection of unknown diseases.

Tack et al. (2020) introduce distribution shifting transformations into SimCLR training. Those are transformations that are so strong that the resulting samples can be considered as OOD and as negatives w.r.t. the original in the SimCLR loss. Similarly to Hendrycks et al. (2019b) and Golan & El-Yaniv (2018), they also train a head that classifies the applied transformation. In a version extended to using in-distribution labels, they consider samples from the same class and with the same transformation as positives, and samples where either is different as negatives. With this method, they obtain OOD detection results that significantly improve over standard classifier confidence, without using any training OOD dataset.

(Liu & Abbeel, 2020) derive a contrastive loss from Joint Energy-Based Model training (Grathwohl et al., 2020) and train it together with cross-entropy on the in-distribution in order to obtain classifiers whose logit values can be transformed into energies that are equivalent as OOD scoring functions to in-distribution density. They show that using these energies yields some improvements over previous density estimation approaches, and that also the classifier confidences show moderately improved OOD detection when compared to standard training.

Liu et al. (2020) propose another Energy-Based method which can incorporate surrogate OOD data during training if such data is available. We analyse this method in detail above, where we show that their Bayes optimal OOD detector is equivalent to the binary discriminator between in- and out-distribution.

(Li & Vasconcelos, 2020) use the same training method as OE and show that careful resampling of the training out-distribution resembling hard negative mining can reduce its size and therefore lead to a more resource effective training OOD dataset, while the resulting models reach similarly good OOD detection performance.

3.6 Experiments

We use CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009) datasets as in-distribution and OpenImages dataset (Krasin et al., 2017) as training out-distribution. We note that the 80 Million Tiny Images (80M) (Torralba et al., 2008) dataset has been used as training out-distribution in the original implementation of Outlier Exposure (Hendrycks et al., 2019a) and several subsequent works that use a surrogate training out-distribution for the CIFAR ID datasets. As a reaction to ethical issues in 80M exposed by Birhane & Prabhu (2021), its authors have retracted the dataset. Because of this development, we use OpenImages as an alternative training out-distribution and only include 80M as an OOD test set, such that its

content do not have any influence on the trained models. Please refer to our published paper (Bitterwolf et al., 2022) for an in-depth discussion of this choice and appended tables with 80M used during training as a reference for comparison with other works that use it as their main surrogate out-distribution. Since we conducted the experimental analysis presented in in Chapter 5 before these issues were raised and 80M was retracted, the experiments there still use 80M.

We use as OOD detection metric the false positive rate at 95% true positive rate, $\text{FPR@95\%TPR}$. For CIFAR-10 and CIFAR-100 evaluate the OOD detection performance on the following datasets: SVHN (Netzer et al., 2011), resized LSUN Classroom (Yu et al., 2015), Uniform Noise, Smooth Noise generated as described by (Hein et al., 2019), the respective other CIFAR dataset, 80M, and CelebA (Liu et al., 2015). We highlight that none of the listed methods has access to those test distributions during training or for fine-tuning as we try to assess the ability of an OOD aware model to generalize to unseen distributions. The FPR for the OpenImages test set is not part of the Mean FPR, since this data *has* been seen during training.

The binary discriminators (BINDISC) as well as the classifiers with background class (BGC) and the shared binary discriminator+classifier (SHARED) of $p(i|x)$ and $p(y|x, i)$ are trained on the 40-2 Wide Residual Network (Zagoruyko & Komodakis, 2016) architecture with the same training schedule as used in (Hendrycks et al., 2019a) for training their Outlier Exposure(OE) models. This includes averaging the loss over batches that are twice as large for the out-distribution. This way we ensure that the differences do not arise due to differences in the training schedules or other important details but only on the employed objectives. In addition to their standard augmentation and normalization, we apply AutoAugment (Cubuk et al., 2019) without Cutout, and we use $\lambda = 1$ where applicable, which from our observations is a sound choice. For the ENERGY OOD detector, we fine-tune the plain model for 10 epochs with OpenImages as out-distribution. For the Mahalanobis OOD detector (Lee et al., 2018b), we use the models and code published by the authors and use OpenImages for the fine tuning of input noise and layer weighting regression. Our code is available at https://github.com/j-cb/Breaking_Down_OOD_Detection.

In addition to the results for CIFAR-10 and CIFAR-100, we also provide results for Restricted ImageNet. This dataset, introduced by Tsipras et al. (2019), consists of 9 classes, where each individual class is a union of multiple ImageNet (Deng et al., 2009) classes, for example the Restricted ImageNet class 'dog' contains all dog breeds from ImageNet. As Restricted ImageNet only contains animal classes, the union over all its classes does not cover the entire ILSVRC2012 dataset (Russakovsky et al., 2015), which allows us to use the remaining ILSVRC2012 classes as training out-distribution "NotRIN". Like we did for the CIFAR experiments, we train a plain classifier, an Outlier Exposure model, a background class model and a shared discriminator/classifier and evaluate them with the different scoring functions. The model is a ResNet50 and we use random cropping and flipping as data augmentation during training. For Restricted ImageNet, we use Flowers (Nilsback & Zisserman, 2008), FGVC Aircraft (Maji et al., 2013), Stanford Cars (Krause et al., 2013) and smooth noise as test out-distributions.

3.6.1 Out-distribution aware training with labeled in-distribution data

In Table 3.1 we compare multiple OOD methods trained with training out-distribution Open-Images and CIFAR-10/100 as in-distribution: confidence of standard training (PLAIN) and OE, MAHALANOBIS detection, ENERGY-Based OOD detection, classifier with background class (BGC) and the combination of a plain classifier and a binary in-vs-out-distribution classifier with shared representation (SHARED COMBI). As described in Section 3.2, both BGC and SHARED COMBI can be used in combination with different scoring functions. For BGC, we evaluate all three scoring functions s_1 , s_2 and s_3 and for SHARED COMBI we only use s_2 and s_3 as s_1 is equivalent to $p(i|x)$ which is the output of SHARED BINDISC. Additionally, we evaluate OOD detection based on the confidence of the shared classifier (SHARED CLASSI) trained together with SHARED BINDISC.

For CIFAR-10, a first interesting observation is that SHARED CLASSI has remarkably good OOD performance; significantly better than a normal classifier (plain) even though it is just trained using normal cross-entropy loss and so the OOD performance is only due to the regularization enforced by the shared representation with SHARED BINDISC. In fact SHARED BINDISC has already good OOD performance with a mean FPR@95%TPR of 19.56, which is improved by considering scoring function s_2/s_3 in the combination of SHARED BINDISC and SHARED CLASSI which yields very good classification accuracy and mean FPR/AUC. Moreover, interesting are the results of the classifier with background class (BGC) which is the method recently advocated in (Thulasidasan et al., 2021). It works very well but the performance depends on the chosen scoring function. Whereas s_1 (output of the background class) is a usable scoring function (mean FPR: 18.83), the maximum probability over the other classes s_2 (mean FPR: 16.52) or the combination in terms of s_3 (mean FPR: 16.63) performs better. In total with the scoring function s_2/s_3 integrating classifier and discriminative information, BGC reaches similar performance to OE (which implicitly also uses s_3 as scoring function). In general, the differences of the methods are relatively minor both in terms of OOD detection and classification accuracy, where the latter is better for all OOD methods compared to the plain classifier; this is most likely explained by better learned representations, see also Hendrycks et al. (2019a); Augustin et al. (2020) for similar observations. The results for CIFAR-100 are similar to CIFAR-10, with some reversals of the overall rankings of the compared methods. OE achieves comparable OOD results to BGC s_2/s_3 and SHARED COMBI s_2/s_3 . For this in-distribution our BGC s_1 and SHARED BINDISC perform best in terms of OOD performance. Classification test accuracy is slightly higher for BGC and SHARED, but the differences are minor.

The above observations are confirmed in the experiments with Restricted ImageNet as in-distribution and the remaining ImageNet classes as training out-distribution. There, we see once more that SHARED CLASSI has relatively good OOD detection performance and clearly beats the plain classifier from standard training. Again, we see that Outlier Exposure (Hendrycks et al., 2019a), training with background class and shared training of classifier and binary discriminator perform similarly. In terms of accuracy, the shared model benefits most from the added unlabelled data compared to plain training. At 95% TPR, the shared

trained binary discriminator and its combinations SHARED COMBI s_2 and SHARED COMBI s_2 detect flower images significantly better than the other approaches which results in the best mean FPR@95%TPR compared to the other methods, while in terms of AUC, OE has a slight advantage.

Overall, as suggested by the theoretical results on the equivalence of the Bayes optimal classifier of OE with the s_3 scoring function of BGC and SHARED COMBI, we observe that even though these methods are derived and in particular trained with quite different objectives, they behave very similar in our experiments. In total we think that this provides a much better understanding where differences of OOD methods are coming from. Regarding the question of which method and scoring function should be used for a given application, the experimental results across datasets and different out-distributions suggest that their difference is minor and there is no clear best choice. However, in 3.4, we describe a potential situation where the s_3 score and in consequence OE is not powerful enough to distinguish in- and out-of-distribution inputs. On the other hand, in cases where the s_1 score is not very informative as training and test out-distributions largely differ, combining it with the classifier confidences is beneficial. This is why for an unknown situation, we recommend BGC or SHARED COMBI with the s_2 scoring function as the safest option. However, it is an open question if there are also situations where s_2 is fundamentally inferior to s_3 .

3.6.2 Shared representation learning for the binary discriminator

As highlighted above the shared training of SHARED CLASSI and SHARED BINDISC and their combination SHARED COMBI with s_2/s_3 as scoring functions yields strong OOD detection and test accuracy among all methods. Here, we evaluate the importance of training the binary discriminator and the plain classifier with a shared representation in comparison to training two entirely separate models PLAIN CLASSI and SEPARATE BINDISC and their combination SEPARATE COMBI with scoring function s_3 . The results for CIFAR-10 and CIFAR-100 can be found in Table 3.2. In total, we see that separate training in particular for CIFAR-100 leads to worse results compared to shared training as expected as the binary discriminator and the classifier cannot benefit from each other. An interesting curiosity is that the combination of the separate classifier with the binary discriminator trained in a shared fashion (PLAIN $\otimes$ SHA DISC) yields almost the same OOD results as SHARED COMBI even though the classifier is significantly worse. Overall, SHARED COMBI performs significantly better when also considering the better classification accuracy which it inherits from SHARED CLASSI.

3.7 Chapter conclusions

We have analyzed different OOD detection methods and have shown that the simple baseline of a binary discriminator between in-and out-distribution is a powerful OOD detection method if trained in a shared fashion with a classifier. We have also revealed the inner mechanism of Outlier Exposure and training with a background class which unexpectedly use a scoring

function which integrates information from $p(i|x)$ and $p(y|x, i)$. In order to be able to formally reason about these relationships, we have introduced a theoretical notion of equivalence of scoring functions that transfers to the OOD detectors obtained by in the asymptotic case where training objectives minimize the Bayes optimal loss in expectation over the respective training data distributions. In our experimental evaluations, no method consistently outperforms all other methods, which indicates that implementation details and training dynamics can mean that different methods will be best in different situations. When comparing two methods which are equivalent by the notion that we defined, we find that they indeed tend to produce similar experimental OOD detection results.

Recent developments

Narasimhan et al. (2023) explore the Bayes optimal predictors for selective classification, i.e. the rejection of ID samples where the model is deemed to likely make an error, besides OOD detection, and also investigate the influence of explicitly or implicitly coupled training for this problem. Fang et al. (2022) introduce assumptions on in- and out-distributions under which OOD detection is not learnable in terms of probably approximately correct error bounds, but argue that in practical cases, meaningful OOD detectors can often be obtained.

Table 3.1: Accuracy on the in-distribution and FPR@95%TPR for different OOD detectors on various test out-distributions. Green color highlights the **best method on each OOD test set** and bold the **best mean**.

IN-DISTRIBUTION: CIFAR-10										
MODEL	ACC.	MEAN	SVHN	LSUN	UNI	SMTH	C-100	80M	CELA	OPENIM
PLAIN CLASSI	95.16	53.01	47.87	50.00	17.51	65.81	60.43	53.44	76.00	63.71
MAHALANOBIS		36.68	20.97	49.00	0.00	0.00	57.21	48.85	80.71	55.13
ENERGY (s_1)	94.13	18.59	10.39	0.00	0.00	0.00	65.60	53.98	0.15	1.44
OE (s_3)	95.06	15.20	9.58	0.00	0.00	0.00	54.05	42.33	0.45	3.46
BGC s_1		18.83	2.36	0.00	0.00	0.00	72.00	56.41	1.04	0.05
BGC s_2	95.21	16.52	7.51	0.00	0.05	2.10	55.16	44.57	6.26	1.65
BGC s_3	95.21	16.63	7.69	0.00	0.07	2.36	55.19	44.67	6.41	1.74
SHARED BINDISC (s_1)		19.56	4.65	0.00	0.00	0.00	77.50	53.93	0.87	0.04
SHARED CLASSI	95.28	29.34	28.00	7.00	2.33	33.04	58.61	47.90	28.54	35.94
SHARED COMBI s_2	95.28	16.00	8.56	0.00	0.00	0.00	58.80	42.79	1.83	0.61
SHARED COMBI s_3	95.28	16.06	9.00	0.00	0.00	0.00	58.68	42.85	1.91	0.66
IN-DISTRIBUTION: CIFAR-100										
MODEL	ACC.	MEAN	SVHN	LSUN	UNI	SMTH	C-10	80M		OPENIM
PLAIN CLASSI	77.16	67.57	75.50	78.33	22.60	70.98	80.55	77.43		80.80
MAHALANOBIS		53.88	54.36	66.00	46.43	0.06	85.39	71.01		74.69
ENERGY (s_1)	73.47	35.91	39.48	0.00	0.00	0.46	92.76	82.80		1.41
OE (s_3)	77.19	35.03	47.36	0.00	0.67	0.08	84.64	77.42		1.28
BGC s_1		31.14	11.58	0.00	0.00	0.00	93.94	81.29		0.07
BGC s_2	77.61	33.32	37.06	0.00	0.00	0.20	84.50	78.17		1.26
BGC s_3	77.61	33.36	37.27	0.00	0.00	0.20	84.51	78.19		1.27
SHARED BINDISC (s_1)		31.86	10.77	0.00	0.00	0.00	95.25	85.11		0.08
SHARED CLASSI	77.35	67.23	71.05	5.00	97.70	69.68	82.05	77.89		28.38
SHARED COMBI s_2	77.35	33.01	37.30	0.00	0.00	1.06	82.71	77.01		1.80
SHARED COMBI s_3	77.35	33.06	37.57	0.00	0.00	1.13	82.68	77.01		1.85
IN-DISTRIBUTION: RESTRICTED IMAGENET										
MODEL	ACC.	MEAN	FLWR	FGVC	CARS	SMTH	UNI			NOTRIN
PLAIN CLASSI	96.34	36.71	60.11	50.23	73.20	0.00	0.00			50.37
OE (s_3)	97.10	4.26	21.06	0.18	0.04	0.00	0.00			6.91
BGC s_1		4.22	20.74	0.33	0.01	0.00	0.00			6.46
BGC s_2	97.50	4.77	23.43	0.39	0.02	0.00	0.00			6.02
BGC s_3	97.50	4.73	23.26	0.36	0.01	0.00	0.00			6.13
SHARED BINDISC (s_1)		2.73	10.10	0.24	0.04	3.26	0.00			5.93
SHARED CLASSI	97.59	17.51	50.54	17.79	19.21	0.00	0.00			21.99
SHARED COMBI s_2	97.59	2.55	12.28	0.45	0.04	0.00	0.00			5.58
SHARED COMBI s_3	97.59	2.62	12.60	0.45	0.04	0.00	0.00			5.63

Table 3.2: Evaluation (same metrics as in Table 3.1) of models trained with shared and separate representations. Shared training benefits both the classifier and the binary discriminators.

IN-DISTRIBUTION: CIFAR-10										
MODEL	ACC.	MEAN FPR	SVHN FPR	LSUN FPR	UNI FPR	SMOOTH FPR	C-100 FPR	80M FPR	CELA FPR	OPENIM FPR
SEPARATE BINDISC (s_1)		23.49	6.21	0.00	0.00	0.00	83.79	65.77	8.68	0.00
PLAIN CLASSI	95.16	53.01	47.87	50.00	17.51	65.81	60.43	53.44	76.00	63.71
SEPARATE COMBI s_3	95.16	21.40	13.15	0.00	0.00	0.00	59.96	49.78	26.93	0.45
SHARED BINDISC (s_1)		19.56	4.65	0.00	0.00	0.00	77.50	53.93	0.87	0.04
SHARED CLASSI	95.28	29.34	28.00	7.00	2.33	33.04	58.61	47.90	28.54	35.94
SHARED COMBI s_3	95.28	16.06	9.00	0.00	0.00	0.00	58.68	42.85	1.91	0.66
PLAIN $\otimes$ SHA DISC s_3	95.16	15.96	8.10	0.00	0.00	0.00	58.48	42.60	2.53	0.70
IN-DISTRIBUTION: CIFAR-100										
MODEL	ACC.	MEAN FPR	SVHN FPR	LSUN FPR	UNI FPR	SMOOTH FPR	C-10 FPR	80M FPR		OPENIM FPR
SEPARATE BINDISC (s_1)		32.50	14.28	0.00	0.00	0.00	96.50	84.22		0.02
PLAIN CLASSI	77.16	67.57	75.50	78.33	22.60	70.98	80.55	77.43		80.80
SEPARATE COMBI s_3	77.16	41.94	69.44	0.00	0.00	24.39	81.15	76.67		0.89
SHARED BINDISC (s_1)		31.86	10.77	0.00	0.00	0.00	95.25	85.11		0.08
SHARED CLASSI	77.35	67.23	71.05	5.00	97.70	69.68	82.05	77.89		28.38
SHARED COMBI s_3	77.35	33.06	37.57	0.00	0.00	1.13	82.68	77.01		1.85
PLAIN $\otimes$ SHA DISC s_3	77.16	33.38	37.00	0.00	0.00	5.42	81.33	76.50		2.23

Part III

Adversarially Robust Out-of-Distribution Detection

Chapter 4

Empirically Adversarially Robust Out-of-Distribution Detection

This chapter is based on the paper “Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem” (Hein et al., 2019) which received an ‘oral presentation’ at CVPR 2019. Matthias Hein came up with the issue of asymptotic overconfidence and the idea of developing a variant of adversarial training for OOD overconfidence and wrote most of the methodological content. Maksym Andriushchenko and I jointly wrote the code for training and evaluating the CEDA and ACET models, and while I trained and evaluated these models, Maksym Andriushchenko conducted the analysis of a two-hidden layer network and the experiments on asymptotic overconfidence.

4.1 Goals and overview

In this chapter, we introduce confidence increasing adversarial attacks on standard models and models that are specifically trained to function as OOD detectors. Besides small sized perturbations, show that basically any classifier whose architecture is part of the class of ReLU networks produces arbitrarily high confidence predictions when moving far enough away from the training data, which works in almost any direction, without an attacking search being necessary. Here, we call ReLU networks any networks consisting of fully connected and convolutional with or without residual connections, where just ReLU or leaky ReLU are used as activation functions and max or average pooling are used in the convolution layers. This means that the class of ReLU networks encompasses those popular neural network architectures which result in a piecewise affine classifier function. The issue of arbitrarily high confidence predictions implies that techniques which operate on the output of the classifier cannot identify these inputs as out-of-distribution inputs. On the contrary we formalize the well known fact that RBF networks produce almost uniform confidence over the classes far away from the training data, which shows that there exist classifiers which satisfy the minimal requirement of not being confident in areas where one has never seen data. Moreover, we propose a robust optimization scheme motivated by adversarial training (Madry et al., 2018) which simply enforces uniform confidence predictions on noise images which are by construction far away from the true images. We show that our technique not only significantly reduces

confidence on such noise images, but also on other unrelated image classification tasks and in some cases even for adversarial samples generated for the original classification task. The training procedure is simple, needs no adaptation for different out-of-distribution tasks, has similar complexity as standard adversarial training and achieves similar performance on the original classification task.

4.2 ReLU networks produce piecewise affine functions

We quickly review in this section the fact that ReLU networks lead to continuous piecewise affine classifiers, see (Arora et al., 2018; Croce & Hein, 2018), which we briefly summarize in order to set the ground for our main theoretical result in Section 4.3.

Definition 2. A function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is called piecewise affine if there exists a finite set of polytopes $\{Q_r\}_{r=1}^M$ (referred to as linear regions of f) such that $\cup_{r=1}^M Q_r = \mathbb{R}^d$ and f is an affine function when restricted to every Q_r .

Feedforward neural networks which use piecewise affine activation functions (e.g. ReLU, leaky ReLU) and are linear in the output layer can be rewritten as continuous piecewise affine functions (Arora et al., 2018). This includes fully connected, convolutional, residual layers and even skip connections as all these layers are just linear mappings. Moreover, it includes further average pooling and max pooling. More precisely, the classifier is a function $f : \mathbb{R}^d \rightarrow \mathbb{R}^K$, where K are the number of classes, such that each component $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$, is a continuous piecewise affine function and the K components $(f_i)_{i=1}^K$ have the same set of linear regions. Note that explicit upper bounds on the number of linear regions have been given (Montufar et al., 2014).

In the following we follow (Croce & Hein, 2018). For simplicity we just present fully connected layers (note that convolutional layers are a particular case of them). Denote by $\sigma : \mathbb{R} \rightarrow \mathbb{R}$, $\sigma(t) = \max\{0, t\}$, the ReLU activation function, by $L + 1$ the number of layers and $W^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ and $b^{(l)} \in \mathbb{R}^{n_l}$ respectively are the weights and offset vectors of layer l , for $l = 1, \dots, L + 1$ and $n_0 = d$. For $x \in \mathbb{R}^d$ one defines $g^{(0)}(x) = x$. Then one can recursively define the pre- and post-activation output of every layer as

$$\begin{aligned} f^{(k)}(x) &= W^{(k)} g^{(k-1)}(x) + b^{(k)}, \quad \text{and} \\ g^{(k)}(x) &= \sigma(f^{(k)}(x)), \quad k = 1, \dots, L, \end{aligned}$$

so that the resulting classifier is obtained as $f^{(L+1)}(x) = W^{(L+1)} g^{(L)}(x) + b^{(L+1)}$.

Let $\Delta^{(l)}, \Sigma^{(l)} \in \mathbb{R}^{n_l \times n_l}$ for $l = 1, \dots, L$ be diagonal matrices defined elementwise as

$$\Delta^{(l)}(x)_{ij} = \begin{cases} \text{sign}(f_i^{(l)}(x)) & \text{if } i = j, \\ 0 & \text{else.} \end{cases},$$

$$\Sigma^{(l)}(x)_{ij} = \begin{cases} 1 & \text{if } i = j \text{ and } f_i^{(l)}(x) > 0, \\ 0 & \text{else.} \end{cases}.$$

Note that for leaky ReLU the entries would be 1 and α instead. This allows to write $f^{(k)}(x)$ as composition of affine functions, that is

$$f^{(k)}(x) = W^{(k)} \Sigma^{(k-1)}(x) \left(W^{(k-1)} \Sigma^{(k-2)}(x) \right. \\ \left. \times \left(\dots \left(W^{(1)} x + b^{(1)} \right) \dots \right) + b^{(k-1)} \right) + b^{(k)},$$

We can further simplify the previous expression as $f^{(k)}(x) = V^{(k)} x + a^{(k)}$, with $V^{(k)} \in \mathbb{R}^{n_k \times d}$ and $a^{(k)} \in \mathbb{R}^{n_k}$ given by

$$V^{(k)} = W^{(k)} \left(\prod_{l=1}^{k-1} \Sigma^{(k-l)}(x) W^{(k-l)} \right) \quad \text{and}$$

$$a^{(k)} = b^{(k)} + \sum_{l=1}^{k-1} \left(\prod_{m=1}^{k-l} W^{(k+1-m)} \Sigma^{(k-m)}(x) \right) b^{(l)}.$$

The polytope $Q(x)$, the linear region containing x , can be characterized as an intersection of $N = \sum_{l=1}^L n_l$ half spaces given by

$$\Gamma_{l,i} = \{z \in \mathbb{R}^d \mid \Delta^{(l)}(x) (V_i^{(l)} z + a_i^{(l)}) \geq 0\},$$

for $l = 1, \dots, L, i = 1, \dots, n_l$, namely

$$Q(x) = \bigcap_{l=1, \dots, L} \bigcap_{i=1, \dots, n_l} \Gamma_{l,i}.$$

Note that N is also the number of hidden units of the network. Finally, we can write

$$f^{(L+1)}(z) \Big|_{Q(x)} = V^{(L+1)} z + a^{(L+1)},$$

which is the affine restriction of f to $Q(x)$.

4.3 Why ReLU networks produce high confidence predictions far away from the training data

With the explicit description of the piecewise linear classifier resulting from a ReLU type network from Section 4.2, we can now formulate our main theorem. It shows that, as long as a very mild condition on the network holds, for any $\varepsilon > 0$ one can always find for (almost) **all** directions an input z far away from the training data which realizes a confidence of $1 - \varepsilon$ on z for a certain class. However, before we come to this result, we first present a technical lemma needed in the proof, which uses that all linear regions are polytopes and thus convex sets.

Lemma 4. *Let $\{Q_i\}_{i=1}^R$ be the set of linear regions associated to the ReLU-classifier $f : \mathbb{R}^d \rightarrow \mathbb{R}^K$. For any $x \in \mathbb{R}^d$ there exists $\alpha \in \mathbb{R}$ with $\alpha > 0$ and $t \in \{1, \dots, R\}$ such that $\beta x \in Q_t$ for all $\beta \geq \alpha$.*

Proof. Suppose the statement would be false. Then there exist $\{\beta_i\}_{i=1}^\infty$ with $\beta_i \geq 0$, $\beta_i \geq \beta_j$ if $i \leq j$ and $\beta_i \rightarrow \infty$ as $i \rightarrow \infty$ such that for $\gamma \in [\beta_i, \beta_{i+1})$ we have $\gamma x \in Q_{r_i}$ with $r_i \in \{1, \dots, R\}$ and $r_{i-1} \neq r_i \neq r_{i+1}$. As there are only finitely many regions there exist $i, j \in \mathbb{N}$ with $i < j$ such that $r_i = r_j$, in particular $\beta_i x \in Q_{r_i}$ and $\beta_j x \in Q_{r_i}$. However, as the linear regions are convex sets also the line segment $[\beta_i x, \beta_j x] \in Q_{r_i}$. However, that implies $\beta_i = \beta_j$ as neighboring segments are in different regions which contradicts the assumption. Thus there can only be finitely many $\{\beta_i\}_{i=1}^M$ and the $\{r_i\}_{i=1}^M$ have to be all different, which finishes the proof. $\square$

Using Lemma 4 we can now state our first main result.

Theorem 4. *Let $\mathbb{R}^d = \cup_{l=1}^R Q_l$ and $f(x) = V^l x + a^l$ be the piecewise affine representation of the output of a ReLU network on Q_l . Suppose that V^l does not contain identical rows for all $l = 1, \dots, R$, then for almost any $x \in \mathbb{R}^d$ and $\varepsilon > 0$ there exists an $\alpha > 0$ and a class $k \in \{1, \dots, K\}$ such that for $z = \alpha x$ it holds*

$$\frac{e^{f_k(z)}}{\sum_{r=1}^K e^{f_r(z)}} \geq 1 - \varepsilon.$$

Moreover, $\lim_{\alpha \rightarrow \infty} \frac{e^{f_k(\alpha x)}}{\sum_{r=1}^K e^{f_r(\alpha x)}} = 1$.

Proof. By Lemma 4 there exists a region Q_t with $t \in \{1, \dots, R\}$ and $\beta > 0$ such that for all $\alpha \geq \beta$ we have $\alpha x \in Q_t$. Let $f(z) = V^t z + a^t$ be the affine form of the ReLU classifier f on Q_t . Let $k^* = \arg \max_k \langle v_k^t, x \rangle$, where v_k^t is the k -th row of V^t . As V^t does not contain identical rows, that is $v_l^t \neq v_m^t$ for $l \neq m$, the maximum is uniquely attained up to a set of measure zero. If the maximum is unique, it holds for sufficiently large $\alpha \geq \beta$

$$\langle v_l^t - v_{k^*}^t, \alpha x \rangle + a_l^t - a_{k^*}^t < 0, \forall l \in \{1, \dots, K\} \setminus \{k^*\}. \quad (4.1)$$

4.3 Why ReLU networks produce high confidence predictions far away from the training data

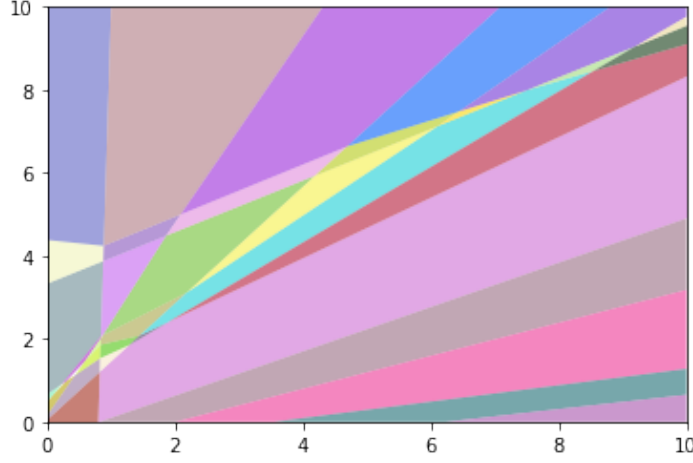


Figure 4.1: A decomposition of $\mathbb{R}^2$ into a finite set of polytopes for a two-hidden layer ReLU network. The outer polytopes extend to infinity. This is where ReLU networks realize arbitrarily high confidence predictions. The picture is produced with the code of (Jordan et al., 2019).

Thus $\alpha x \in Q_t$ is classified as k^* . Moreover,

$$\frac{e^{f_{k^*}(\alpha x)}}{\sum_{l=1}^K e^{f_l(\alpha x)}} = \frac{e^{\langle v_{k^*}^t, \alpha x \rangle + a_k^t}}{\sum_{l=1}^K e^{\langle v_l^t, \alpha x \rangle + a_l^t}} \quad (4.2)$$

$$= \frac{1}{1 + \sum_{l \neq k^*}^K e^{\langle v_l^t - v_{k^*}^t, \alpha x \rangle + a_l^t - a_k^t}}. \quad (4.3)$$

By inequality (4.1) all the terms in the exponential are negative and thus by upscaling α , using $\langle v_{k^*}^t, x \rangle > \langle v_l^t, x \rangle$ for all $l \neq k^*$, we can get the exponential term arbitrarily close to 0. In particular,

$$\lim_{\alpha \rightarrow \infty} \frac{1}{1 + \sum_{l \neq k^*}^K e^{\langle v_l^t - v_{k^*}^t, \alpha x \rangle + a_l^t - a_k^t}} = 1.$$

□

Please note that the condition that for a region the linear part V^l need not contain two identical rows is very weak. It is hardly imaginable that this is ever true for a normally trained network unless the output of the network is constant anyway. Even if it is true, it just invalidates the assertion of the theorem for the points lying in this region. Without explicitly enforcing this condition it seems impossible that this is true for all possible asymptotic regions extending to infinity (see Figure 4.1). However, it is also completely open how this condition could be enforced during training of the network.

The result implies that for ReLU networks there exist infinitely many inputs which realize arbitrarily high confidence predictions of the networks. It is easy to see that the temperature rescaling of the softmax, $\frac{e^{f_k(x)/T}}{\sum_{l=1}^K e^{f_l(x)/T}}$, for temperature $T > 0$, as used in (Liang et al., 2018), will not be able to detect these cases, in particular since the first step of the method in (Liang et al., 2018) consists of going in the direction of increasing confidence. Also it is obvious that using a reject option in the classifier, see e.g. (Bartlett & Wegkamp, 2008), will not help to detect these instances either. The result is negative in the sense that it looks like that without modifying the architecture of a ReLU network it is impossible to prevent this phenomenon. Please note that from the point of view of Bayesian decision theory the softmax function is the correct transfer function (Lapin et al., 2016) for the cross-entropy loss turning the classifier output $f_k(x)$ into an estimate $P(Y = k | x, f) = \frac{e^{f_k(x)}}{\sum_{l=1}^K e^{f_l(x)}}$ for the conditional probability at x .

While the previous result seems not to be known, the following result is at least qualitatively known (Goodfellow et al., 2015) but we could not find a reference for it. In contrast to the ReLU networks it turns out that Radial Basis Function (RBF) networks have the property to produce approximately uniform confidence predictions far away from the training data. Thus there exist classifiers which satisfy the minimal requirement which we formulated in Section 4.1. In the following theorem we explicitly quantify what “far away” means in terms of parameters of the RBF classifier and the training data.

Theorem 5. Let $f_k(x) = \sum_{l=1}^N \alpha_{kl} e^{-\gamma \|x - x_l\|_2^2}$, $k = 1, \dots, K$ be an RBF-network trained with cross-entropy loss on the training data $(x_i, y_i)_{i=1}^N$. We define $r_{\min} = \min_{l=1, \dots, N} \|x - x_l\|_2$ and $\alpha = \max_{r,k} \sum_{l=1}^N |\alpha_{rl} - \alpha_{kl}|$. If $\varepsilon > 0$ and

$$r_{\min}^2 \geq \frac{1}{\gamma} \log \left(\frac{\alpha}{\log(1 + K\varepsilon)} \right),$$

then for all $k = 1, \dots, K$,

$$\frac{1}{K} - \varepsilon \leq \frac{e^{f_k(x)}}{\sum_{r=1}^K e^{f_r(x)}} \leq \frac{1}{K} + \varepsilon.$$

Proof. It holds $\frac{e^{f_k(x)}}{\sum_{r=1}^K e^{f_r(x)}} = \frac{1}{\sum_{r=1}^K e^{f_r(x) - f_k(x)}}$. With

$$|f_r(x) - f_k(x)| = \left| \sum_l (\alpha_{rl} - \alpha_{kl}) e^{-\gamma \|x - x_l\|_2^2} \right| \quad (4.4)$$

$$\leq e^{-\gamma r_{\min}^2} \sum_l |\alpha_{rl} - \alpha_{kl}| \quad (4.5)$$

$$\leq e^{-\gamma r_{\min}^2} \alpha \leq \log(1 + K\varepsilon), \quad (4.6)$$

where the last inequality follows by the condition on $r_{\min}$. We get

$$\frac{1}{\sum_{r=1}^K e^{f_r(x)-f_k(x)}} \geq \frac{1}{\sum_{r=1}^K e^{|f_r(x)-f_k(x)|}} \quad (4.7)$$

$$\geq \frac{1}{Ke^{\alpha e^{-r_{\min}^2}}} \quad (4.8)$$

$$\geq \frac{1}{K} \frac{1}{1+K\varepsilon} \geq \frac{1}{K} - \varepsilon, \quad (4.9)$$

where we have used in the third inequality the condition on $r_{\min}^2$ and in the last step we use $1 \geq (1 - K\varepsilon)(1 + K\varepsilon) = 1 - K^2\varepsilon^2$. Similarly, we get

$$\begin{aligned} \frac{1}{\sum_{r=1}^K e^{f_r(x)-f_k(x)}} &\leq \frac{1}{\sum_{r=1}^K e^{-|f_r(x)-f_k(x)|}} \\ &\leq \frac{1}{Ke^{-\alpha e^{-r_{\min}^2}}} \\ &\leq \frac{1}{K}(1 + K\varepsilon) \leq \frac{1}{K} + \varepsilon. \end{aligned}$$

This finishes the proof. $\square$

We think that it is a very important open problem to realize a similar result as in Theorem 5 for a class of neural networks. Note that arbitrarily high confidence predictions for ReLU networks can be obtained only if the domain is unbounded, e.g. $\mathbb{R}^d$. However, images are contained in $[0, 1]^d$ and thus Theorem 4 does not directly apply, even though the technique can in principle be used to produce high-confidence predictions (see Table 4.3). In the next section we propose a novel training scheme enforcing low confidence predictions on inputs far away from the training data.

4.4 ACET: Adversarial Confidence Enhanced Training

In this section we suggest a simple way to adjust the confidence estimation of a neural network far away from the training data, not necessarily restricted to ReLU networks studied in Theorem 4. Theorem 4 tells us that for ReLU networks a post-processing of the softmax scores is not sufficient to avoid high-confidence predictions far away from the training data - instead there seem to be two potential ways to tackle the problem: a) one uses an extra generative model either for the in-distribution or for the out-distribution or b) one modifies directly the network via an adaptation of the training process so that uniform confidence predictions are enforced far away from the training data. As recently problems with generative models have been pointed out which assign high confidence to samples from the out-distribution (Nalisnick et al., 2019) and thus a) seems less promising, we explore approach b).

We assume that it is possible to characterize a distribution of data points p_{out} on the input space for which we are sure that they do not belong to the true distribution p_{in} resp. the set of the intersection of their supports has zero or close to zero probability mass. An example of such an out-distribution p_{out} would be the uniform distribution on $[0, 1]^{w \times h}$ ($w \times h$ gray scale images) or similar noise distributions. Suppose that the in-distribution consists of certain image classes like handwritten digits, then the probability mass of all images of handwritten digits under the p_{out} is zero (if it is really a low-dimensional manifold) or close to zero.

In such a setting the training objective can be written as a sum of two losses:

$$\frac{1}{N} \sum_{i=1}^N L_{CE}(y_i, f(x_i)) + \lambda \mathbb{E}[L_{p_{\text{out}}}(f, Z)], \quad (4.10)$$

where $(x_i, y_i)_{i=1}^N$ is the i.i.d. training data, Z has distribution p_{out} and

$$L_{CE}(y_i, f(x_i)) = -\log \left(\frac{e^{f_{y_i}(x_i)}}{\sum_{k=1}^K e^{f_k(x_i)}} \right) \quad (4.11)$$

$$L_{p_{\text{out}}}(f, z) = \max_{l=1, \dots, K} \log \left(\frac{e^{f_l(z)}}{\sum_{k=1}^K e^{f_k(z)}} \right). \quad (4.12)$$

L_{CE} is the usual cross entropy loss on the original classification task and $L_{p_{\text{out}}}(f, z)$ is the maximal log confidence over all classes, where the confidence of class l is given by $\frac{e^{f_l(z)}}{\sum_{k=1}^K e^{f_k(z)}}$, with the softmax function as the link function. The full loss can be easily minimized by using SGD with batchsize B for the original data and adding $\lceil \lambda B \rceil$ samples from p_{out} on which one enforces a uniform distribution over the labels. We call this process in the following *confidence enhancing data augmentation (CEDA)*. We note that in a concurrent paper (Hendrycks et al., 2019a) a similar scheme has been proposed, where they use as p_{out} existing large image datasets, whereas we favor an agnostic approach where p_{out} models a certain “noise” distribution on images.

The problem with CEDA is that it might take too many samples to enforce low confidence on the whole out-distribution. Moreover, it has been shown in the area of adversarial manipulation that data augmentation is not sufficient for robust models and we will see in Section 4.5 that indeed CEDA models still produce high confidence predictions in a neighborhood of noise images. Thus, we propose to use ideas from robust optimization similar to adversarial training which (Szegedy et al., 2014; Goodfellow et al., 2015; Madry et al., 2018) apply to obtain robust networks against adversarial manipulations. Thus we are enforcing low confidence not only at the point itself but actively minimize the worst case in a neighborhood of the point. This leads to the following formulation of *adversarial confidence enhancing training (ACET)*

$$\frac{1}{N} \sum_{i=1}^N L_{CE}(y_i, f(x_i)) + \lambda \mathbb{E} \left[\max_{\|u-Z\|_p \leq \epsilon} L_{p_{\text{out}}}(f, u) \right], \quad (4.13)$$

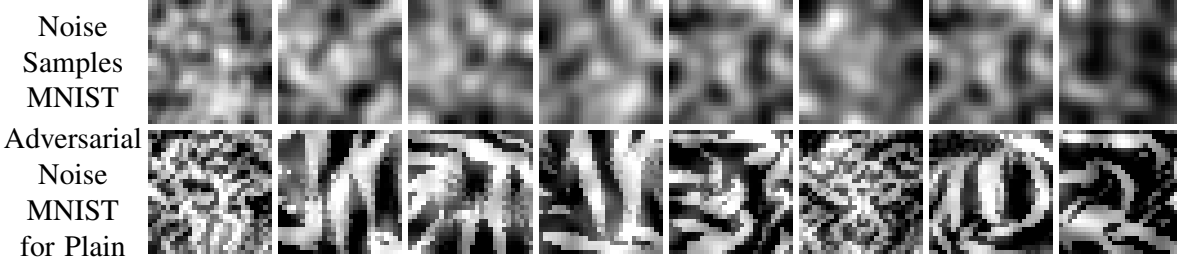


Figure 4.2: Top row: our generated noise images based on uniform noise resp. permuted MNIST together with a Gaussian filter and contrast rescaling. Bottom row: for each noise image from above we generate the corresponding adversarial noise image using PGD with 40 iterations maximizing the second part of the loss in ACET for the plain model. Note that neither in the noise images nor in the adversarially modified ones there is structure similar to a MNIST image. For ACET and CEDA it is very difficult to generate adversarial noise images for the fully trained models thus we omit them.

where in each SGD step one solves (approximately) for a given $z \sim p_{\text{out}}$ the optimization problem:

$$\max_{\|u-z\|_p \leq \varepsilon} L_{p_{\text{out}}}(f, u). \quad (4.14)$$

In this paper we use always $p = \infty$. Note that if the distributions p_{out} and p_{in} have joint support, the maximum in (4.14) could be obtained at a point in the support of the true distribution. However, if p_{out} is a generic noise distribution like uniform noise or a smoothed version of it, then the number of cases where this happens has probability mass close to zero under p_{out} and thus does not negatively influence in (4.13) the loss L_{CE} on the true distribution. The optimization of ACET in (4.13) can be done using an adapted version of the PGD method of (Madry et al., 2018) for adversarial training where one performs projected gradient descent (potentially for a few restarts) and uses the u realizing the worst loss for computing the gradient. The resulting samples are more informative and thus lead to a faster and more significant reduction of high confidence predictions far away from the training data. We use $\varepsilon = 0.3$ for all datasets. We present in Figure 4.2 and 4.3 for MNIST and CIFAR-10 a few noise images together with their adversarial modification u generated by applying PGD to solve (4.14). One can observe that the generated images have no structure resembling images from the in-distribution.

4.5 Experiments

In the evaluation, we follow (Hendrycks & Gimpel, 2017; Liang et al., 2018; Lee et al., 2018a) by training on one dataset and evaluating the confidence on other out of distribution datasets and noise images. In contrast to (Liang et al., 2018; Lee et al., 2018a) we neither use a different parameter set for each test dataset (Liang et al., 2018) nor do we use one of the test datasets during training (Lee et al., 2018a). More precisely, we train on MNIST, SVHN, CIFAR-10 and

Trained on MNIST	Plain (TE: 0.51%)			CEDA (TE: 0.74%)			ACET (TE: 0.66%)		
	MMC	AUC	FPR@95	MMC	AUC	FPR@95	MMC	AUC	FPR@95
MNIST	0.991	–	–	0.987	–	–	0.986	–	–
FMNIST	0.654	0.972	0.121	0.373	0.994	0.027	0.239	0.998	0.003
EMNIST	0.821	0.883	0.374	0.787	0.895	0.358	0.752	0.912	0.313
grayCIFAR-10	0.492	0.996	0.003	0.105	1.000	0.000	0.101	1.000	0.000
Noise	0.463	0.998	0.000	0.100	1.000	0.000	0.100	1.000	0.000
Adv. Noise	1.000	0.031	1.000	0.102	0.998	0.002	0.162	0.992	0.042
Adv. Samples	0.999	0.358	0.992	0.987	0.549	0.953	0.854	0.692	0.782
Trained on SVHN	Plain (TE: 3.53%)			CEDA (TE: 3.50%)			ACET (TE: 3.52%)		
	MMC	AUC	FPR@95	MMC	AC	FPR@95	MMC	AUC	FPR@95
SVHN	0.980	–	–	0.977	–	–	0.978	–	–
CIFAR-10	0.732	0.938	0.348	0.551	0.960	0.209	0.435	0.973	0.140
CIFAR-100	0.730	0.935	0.350	0.527	0.959	0.205	0.414	0.971	0.139
LSUN CR	0.722	0.945	0.324	0.364	0.984	0.084	0.148	0.997	0.012
Imagenet-	0.725	0.939	0.340	0.574	0.955	0.232	0.368	0.977	0.113
Noise	0.720	0.943	0.325	0.100	1.000	0.000	0.100	1.000	0.000
Adv. Noise	1.000	0.004	1.000	0.946	0.062	0.940	0.101	1.000	0.000
Adv. Samples	1.000	0.004	1.000	0.995	0.009	0.994	0.369	0.778	0.279

Table 4.1: On the datasets **MNIST** and **SVHN**, (CIFAR-10 and CIFAR-100 see below), we train three models: Plain, CEDA and ACET. We evaluate them on out-of-distribution samples (other image datasets, noise, adversarial noise and adversarial samples built from the test set on which was trained). We report test error of all models and show the mean maximum confidence (MMC) on the in- and out-distribution samples (lower is better for out-distribution samples), the AUC for the discrimination between in- and out-distribution based on confidence value (higher is better), and the FPR at 95% true positive rate for the same problem (lower is better).

Trained on CIFAR-10	Plain (TE: 8.87%)			CEDA (TE: 8.87%)			ACET (TE: 8.44%)		
	MMC	AUC	FPR@95	MMC	AUC	FPR@95	MMC	AUC	FPR@95
CIFAR-10	0.949	–	–	0.946	–	–	0.948	–	–
SVHN	0.800	0.850	0.783	0.327	0.978	0.146	0.263	0.981	0.118
CIFAR-100	0.764	0.856	0.715	0.761	0.850	0.720	0.764	0.852	0.711
LSUN CR	0.738	0.872	0.667	0.735	0.864	0.680	0.745	0.858	0.677
Imagenet-	0.757	0.858	0.698	0.749	0.853	0.704	0.744	0.859	0.678
Noise	0.825	0.827	0.818	0.100	1.000	0.000	0.100	1.000	0.000
Adv. Noise	1.000	0.035	1.000	0.985	0.032	0.983	0.112	0.999	0.008
Adv. Samples	1.000	0.034	1.000	1.000	0.014	1.000	0.633	0.512	0.590

Trained on CIFAR-100	Plain (TE: 31.97%)			CEDA (TE: 32.74%)			ACET (TE: 32.24%)		
	MMC	AUC	FPR@95	MMC	AUC	FPR@95	MMC	AUC	FPR@95
CIFAR-100	0.751	–	–	0.734	–	–	0.728	–	–
SVHN	0.570	0.710	0.865	0.290	0.874	0.410	0.234	0.912	0.345
CIFAR-10	0.560	0.718	0.856	0.547	0.711	0.855	0.530	0.720	0.860
LSUN CR	0.592	0.690	0.887	0.581	0.678	0.887	0.554	0.698	0.881
Imagenet-	0.531	0.744	0.827	0.504	0.749	0.808	0.492	0.752	0.819
Noise	0.614	0.672	0.928	0.010	1.000	0.000	0.010	1.000	0.000
Adv. Noise	1.000	0.000	1.000	0.985	0.015	0.985	0.013	0.998	0.003
Adv. Samples	0.999	0.010	1.000	0.999	0.012	1.000	0.863	0.267	0.975

Table 4.2: On the datasets **CIFAR-10** and **CIFAR-100**, we train three models: Plain, CEDA and ACET. We evaluate them on out-of-distribution samples (other image datasets, noise, adversarial noise and adversarial samples built from the test set on which was trained). We report test error of all models and show the mean maximum confidence (MMC) on the in- and out-distribution samples (lower is better for out-distribution samples), the AUC for the discrimination between in- and out-distribution based on confidence value (higher is better), and the FPR at 95% true positive rate for the same problem (lower is better).

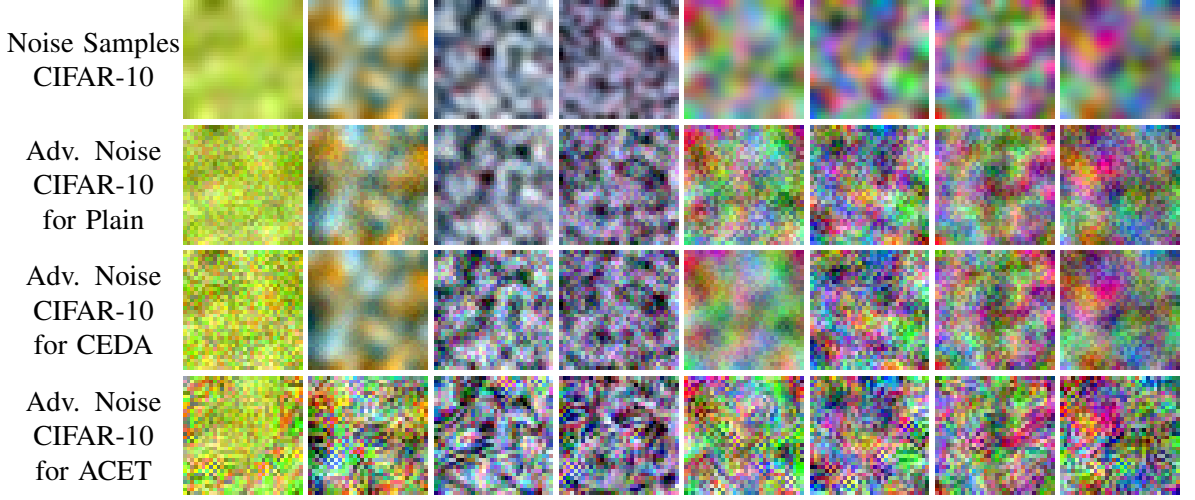


Figure 4.3: Top row: our generated noise images based on uniform noise resp. permuted CIFAR-10 together with a Gaussian filter and contrast rescaling (similar to Figure 4.2). Bottom rows: the corresponding adversarial images for the plain, CEDA, and ACET models. Neither the noise nor the adversarial noise images show similarity to CIFAR-10 images.

CIFAR-100, where we use the LeNet architecture on MNIST taken from (Madry et al., 2018) and a ResNet architecture (He et al., 2016) for the other datasets. We also use standard data augmentation which includes random crops for all datasets and random mirroring for CIFAR-10 and CIFAR-100. For the generation of out-of-distribution images from p_{out} we proceed as follows: half of the images are generated by randomly permuting pixels of images from the training set and half of the images are generated uniformly at random. Then we apply to these images a Gaussian filter with standard deviation $\sigma \in [1.0, 2.5]$ as lowpass filter to have more low-frequency structure in the noise. As the Gaussian filter leads to a contrast reduction we apply afterwards a global rescaling so that the maximal range of the image is again in $[0, 1]$.

4.5.1 Training

We train each model normally (plain), with confidence enhancing data augmentation (CEDA) and with adversarial confidence enhancing training (ACET). It is well known that weight decay alone reduces overconfident predictions. Thus we use weight decay with regularization parameter $5 \cdot 10^{-4}$ for all models leading to a strong baseline (plain). For both CEDA (4.10) and ACET (4.13) we use $\lambda = 1$, that means 50% of the samples in each batch are from the original training set and 50% are noise samples as described before. For ACET we use $p = \infty$ and $\varepsilon = 0.3$ and optimize with PGD (Madry et al., 2018) using 40 iterations and stepsize 0.0075 for all datasets. All models are trained for 100 epochs with ADAM (Kingma & Ba, 2015) on MNIST and SGD+momentum for SVHN/CIFAR-10/CIFAR-100. The initial learning rate is 10^{-3} for MNIST and 0.1 for SVHN/CIFAR-10 and it is reduced by a factor of 10 at the 50th, 75th and 90th of the in total 100 epochs. The code is available at

Table 4.3: **First row:** We evaluate all trained models on uniform random inputs scaled by a constant $\alpha \geq 1$ (note that the resulting inputs will not constitute valid images anymore, since in most cases they exceed the $[0, 1]^d$ box). We find the minimum α such that the models output 99.9% confidence on them, and report the median over 10 000 trials. As predicted by Theorem 4 we observe that it is always possible to obtain overconfident predictions just by scaling inputs by some constant α , and for plain models this constant is smaller than for ACET. **Second row:** we show the percentage of overconfident predictions (higher than 95% confidence) when projecting back the α -rescaled uniform noise images back to $[0, 1]^d$. One observes that there are much less overconfident predictions for ACET compared to standard training.

	Plain				ACET			
	MNIST	SVHN	CIFAR-10	CIFAR-100	MNIST	SVHN	CIFAR-10	CIFAR-100
Median α	1.5	28.1	8.1	9.9	$3.0 \cdot 10^{15}$	49.8	45.3	9.9
% overconf.	98.7%	99.9%	99.9%	99.8%	0.0%	50.2%	3.4%	0.0%

https://github.com/max-andr/relu_networks_overconfident.

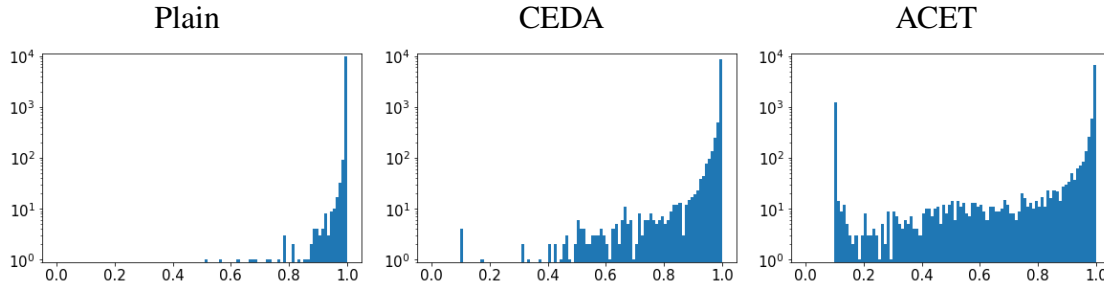


Figure 4.4: Histogram of confidence values (logarithmic scale) of adversarial samples based on MNIST test points. ACET is the only model where a significant fraction of adversarial samples have very low confidence. Note, however that the ACET model has not been trained on adversarial samples of MNIST, but only on adversarial noise.

4.5.2 Evaluation

We report for each model (plain, CEDA, ACET) the test error and the mean maximal confidence (for each point this is $\max_{k=1, \dots, K} \frac{e^{f_k(x)}}{\sum_{l=1}^K e^{f_l(x)}}$), denoted as MMC, on the test set. In order to evaluate how well we reduce the confidence on the out-distribution, we use four datasets on CIFAR-10 (Krizhevsky & Hinton, 2009) and SVHN (Netzer et al., 2011) (namely among

CIFAR-10, CIFAR-100, SVHN, ImageNet-, which is a subset of ImageNet where we removed classes similar to CIFAR-10, and the classroom subset of LSUN (Yu et al., 2015) we use the ones on which we have *not* trained) and for MNIST we evaluate on EMNIST (Cohen et al., 2017), a grayscale version of CIFAR-10 and Fashion MNIST (Xiao et al., 2017). Additionally, we show the evaluation on noise, adversarial noise and adversarial samples. The noise is generated in the same way as the noise we use for training. For adversarial noise, where we maximize the maximal confidence over all classes (see $L_{p_{\text{out}}}(f, z)$ in (4.12)), we use PGD with 200 iterations and stepsize 0.0075 in the ε ball wrt the $\|\cdot\|_{\infty}$ -norm with $\varepsilon = 0.3$ (same as in training). Note that for training we use only 40 iterations, so that the attack at test time is significantly stronger. Finally, we check also the confidence on adversarial samples computed for the test set of the in-distribution dataset using 80 iterations of PGD with $\varepsilon = 0.3$, stepsize 0.0075 for MNIST and $\varepsilon = 0.1$, stepsize 0.0025 for the other datasets. The latter two evaluation modalities are novel compared to (Hendrycks & Gimpel, 2017; Liang et al., 2018; Lee et al., 2018a). The adversarial noise is interesting as it actively searches for images which still yield high confidence in a neighborhood of a noise image and thus is a much more challenging than the pure evaluation on noise. Moreover, it potentially detects an over-adaptation to the noise model used during training in particular in CEDA. The evaluation on adversarial samples is interesting as one can hope that the reduction of the confidence for out-of-distribution images also reduces the confidence of adversarial samples as typically adversarial samples are off the data manifold (Stutz et al., 2019) and thus are also out-of-distribution samples (even though their distance to the true distribution is small). Note that our models have never seen adversarial samples during training, they only have been trained using the adversarial noise. Nevertheless our ACET model can reduce the confidence on adversarial samples. As evaluation criteria we use the mean maximal confidence, the area under the ROC curve (AUC) where we use the confidence as a threshold for the detection problem (in-distribution vs. out-distribution). Moreover, we report in the same setting the false positive rate (FPR) when the true positive rate (TPR) is fixed to 95%.

4.5.3 Main results

In Tables 4.1&4.2, we show the results of plain (normal training), CEDA and ACET. First of all, we observe that there is almost no difference between the test errors of all three methods. Thus improving the confidence far away from the training data does not impair the generalization performance. We also see that the plain models always produce relatively high confidence predictions on noise images and completely fail on adversarial noise. CEDA produces low confidence on noise images but mostly fails (except for MNIST) on adversarial noise which was to be expected as similar findings have been made for the creation of adversarial samples. Only ACET consistently produces low confidence predictions on adversarial noise and has high AUROC. For the out-of-distribution datasets, CEDA and ACET improve most of the time the maximal confidence and the AUROC, sometimes with very strong improvements like on MNIST evaluated on FMNIST or SVHN evaluated on LSUN. However, one observes that it is more difficult to reduce the confidence for related tasks e.g. MNIST evaluated on

EMNIST or CIFAR-10 evaluated on LSUN, where the image structure is more similar. Finally, an interesting outcome is that ACET reduces the confidence on adversarial examples, see Figure 4.4 for an illustration for MNIST, and achieves on all datasets improved AUROC values so that one can detect more adversarial examples via thresholding the confidence compared to the plain and CEDA models. The improved performance of ACET is to some extent unexpected as we just bias the model towards uniform confidence over all classes far away from the training data, but adversarial examples are still close to the original images. In summary, ACET does improve confidence estimates significantly compared to the plain model but also compared to CEDA, in particular on adversarial noise and adversarial examples. ACET has also a beneficial effect on adversarial examples which is an interesting side effect and shows in our opinion that the models have become more reliable.

4.5.4 Far away high confidence predictions

Theorem 4 states that ReLU networks always attain high confidence predictions far away from the training data. The two network architectures used in this paper are ReLU networks. It is thus interesting to investigate if the confidence-enhanced training, ACET, makes it harder to reach high confidence than for the plain model. We do the following experiment: we take uniform random noise images x and then search for the smallest α such that the classifier attains 99.9% confidence on αx . This is exactly the construction from Theorem 4 and the result can be found in Table 4.3.

We observe that indeed the required upscaling factor α is significantly higher for ACET than for the plain models which implies that our method also influences the network far away from the training data. This also shows that even training methods explicitly aiming at counteracting the phenomenon of high confidence predictions far away from the training data, cannot prevent this.

We also do a similar α -scaling experiment, but with the projection to the image domain ($[0, 1]^d$ box), and report the percentage of overconfident predictions (higher than 95% confidence) in Table 4.3, second row. We observe that such a technique can lead to overconfident predictions even in the image domain for the plain models. At the same time, on all datasets, the ACET models have a significantly smaller fraction of overconfident examples compared to the plain models.

4.5.5 Image level effects of ACET on ID and OOD confidences

We compare predictions of the plain model trained on MNIST (Figure 4.5) and the model trained with ACET (Figure 4.6). We analyze the images that receive the lowest maximum confidence on the original dataset (MNIST), and the highest maximum confidence on the two datasets that were used for evaluation (EMNIST, grayCIFAR-10).

On the MNIST in-distribution, we observe that for both models the lowest maximum confidence corresponds to hard input images that are either discontinuous, rotated or simply ambiguous.

On EMNIST, we note that some handwritten letters like 'o' and 'i' may look exactly the same as digits '0' and '1'. Therefore, one should not expect that an ideal model assigns uniform confidences to all EMNIST images. For Figure 4.5 and Figure 4.6 we consider predictions on letters that in general do not look exactly like digits ('a', 'b', 'c', 'd'). We observe that the images with the highest maximum confidence correspond to the handwritten letters that *resemble* digits, so the predictions of both models are justified.

Grayscale CIFAR-10 consists of images that are clearly distinct from digits. Thus, one can expect uniform confidences on such images, which is achieved by the ACET model (Tables 4.1&4.2), but not with the plain model. The mean maximum confidence of the ACET model is close to 10%, with several individual images that are scored with up to 40.41% confidence. Note, that this is much better than for the plain model, which assigns up to 99.60% confidence for the images that have nothing to do with digits. This result is particularly interesting, since the ACET model has not been trained on grayCIFAR-10 examples, and yet it shows much better confidence calibration for out-of-distribution samples.

4.6 Chapter conclusions

In this chapter, we introduced and highlighted the problem of classifier overconfidence far away from the training data. We showed and empirically confirmed the theoretical inevitability of reaching arbitrarily high confidence when a standard classifier receives inputs that are very far away from standard images. While the theoretical, asymptotic conclusion may require leaving the image domain, we found that for real classifiers, high confidence already sets in with proper inputs that lie inside or on the edge of the image box. Even on standard OOD inputs from other datasets—and thus other semantic classes—than the training distribution, standard classifiers are often overconfident. We devised CEDA training which alleviates some of these issues by enforcing low confidence on synthetic OOD examples during training. Besides investigating the asymptotic overconfidence issue, we also developed confidence increasing adversarial attacks that slightly modify an OOD input that originally receives low confidence towards a perceptually similar image where the model has close to 100% confidence. The plain and even CEDA models stand basically no chance against these attacks. We therefore introduced ACET, which is trained to have low confidence even on such perturbed inputs. Besides being robust against adversarially perturbed inputs, ACET's confidence increases much slower when moving very far outside of the image domain, and when constraining movement away from the data to the image box, in most cases ACET does not become overconfident.

Recent developments

Fort (2022) highlights that similar attacks are also successful against OOD detectors that are not directly based on confidence, but e.g. on feature-space Mahalanobis distance. Azizmalayeri et al. (2022) use GAN-based ((Goodfellow et al., 2014)) generations of synthetic outliers to train an adversarially robust OOD detector. Rashid et al. (2024) introduce an extra class which by construction of their network is guaranteed to dominate the other classes when moving far away from the training data.

We will discuss various further aspects and improvements of adversarially robust OOD detection that were published since we presented this work—both from ourselves and other authors—in the following chapters.

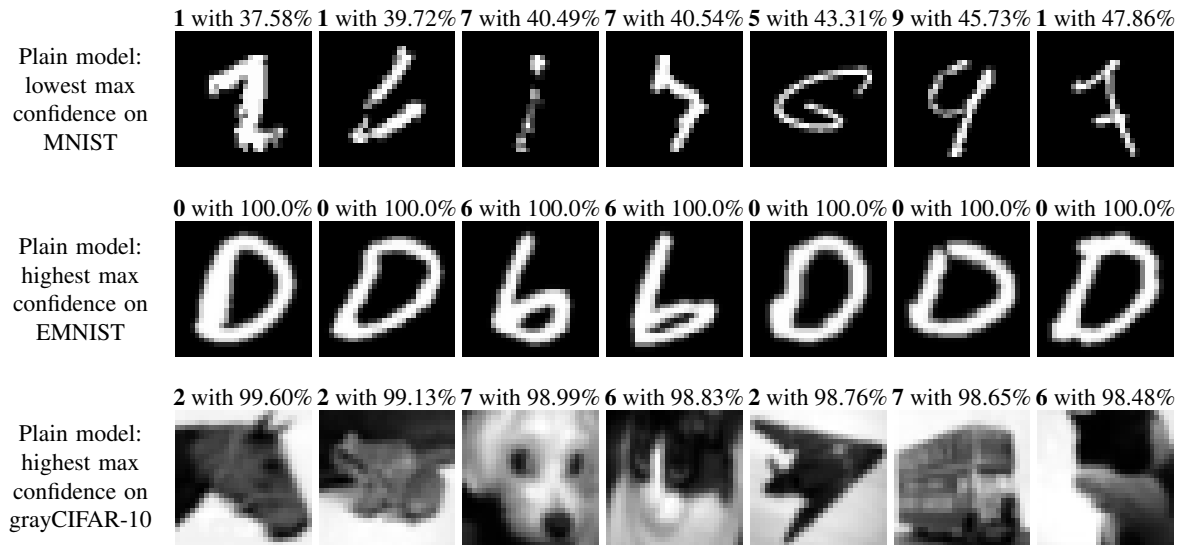


Figure 4.5: Top Row: predictions of the plain MNIST model with the lowest maximum confidence. Middle Row: predictions of the plain MNIST model on letters 'a', 'b', 'c', 'd' of EMNIST with the highest maximum confidence. Bottom Row: predictions of the plain MNIST model on the grayscale version of CIFAR-10 with the highest maximum confidence. Note that although the predictions on EMNIST are mostly justified, the predictions on CIFAR-10 are overconfident on the images that have no resemblance to digits.

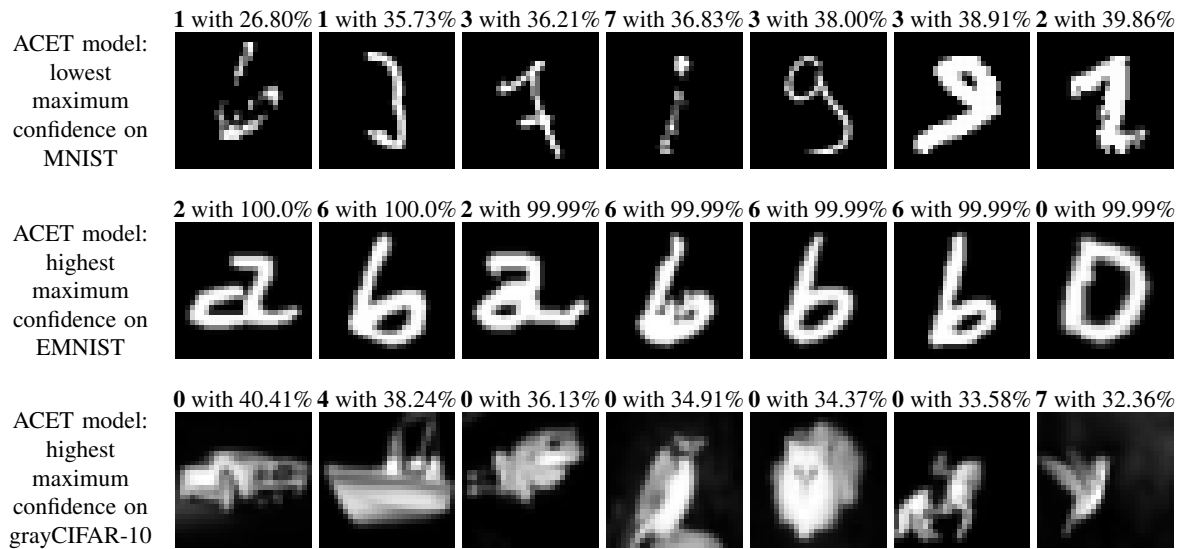


Figure 4.6: Top Row: predictions of the ACET MNIST model with the lowest maximum confidence. Middle Row: predictions of the ACET MNIST model on letters 'a', 'b', 'c', 'd' of EMNIST with the highest maximum confidence. Bottom Row: predictions of the ACET MNIST model on the grayscale version of CIFAR-10 with the highest maximum confidence. Note that for the ACET model the predictions on both EMNIST and grayCIFAR-10 are now justified.

Chapter 5

Certifiably Adversarially Robust Out-of-Distribution Detection

This chapter is based on the paper “Certifiably Adversarially Robust Detection of Out-of-Distribution Data” (Bitterwolf et al., 2020) which we presented at NeurIPS 2020. Alexander Meinke, Matthias Hein and I independently came up with different aspects of the ideas of adversarially certifying OOD detection and modifying IBP to achieve this. I implemented, stabilized and ran the training of GOOD models. Alexander Meinke and Matthias Hein contributed many ideas towards overcoming diversely faceted tendencies of collapse during training of models with provable low confidence on OOD data. Matthias Hein and I developed the idea of using quantile based losses. I was the main author of the paper and Alexander Meinke and Matthias Hein assisted in many parts of the writing. Alexander Meinke improved and ran the adversarial attacks for the empirical robustness evaluations and contributed the detailed analysis of EMNIST as out-distribution for MNIST.

5.1 Motivation for worst-case OOD detection guarantees

In the previous chapter, we demonstrated that it is possible to construct OOD detectors which empirically are robust against confidence increasing adversarial attacks on OOD samples. For some safety critical applications, however, empirical checks are not sufficient in order to trust a deep learning system in a high-stakes decision. Thus provable guarantees on the behavior of a deep learning system are needed.

In the introduction, we already discussed methods that provide provable adversarial robustness for image classifiers. However, in the area of certified out-of-distribution detection the only robustness guarantees for OOD were given in (Meinke & Hein, 2020), where a density estimator for in- and out-distribution is integrated into the predictive uncertainty of the neural network, which allows them to guarantee that far away from the training data the confidence of the neural network becomes uniform over the classes. They provide guarantees on the maximal confidence attained on l_2 -type balls around uniform noise. However, this technique is not able to provide meaningful guarantees around points which are similar or even close to the in-distribution data and, as we will show, provide only weak guarantees against l_∞ -adversaries.

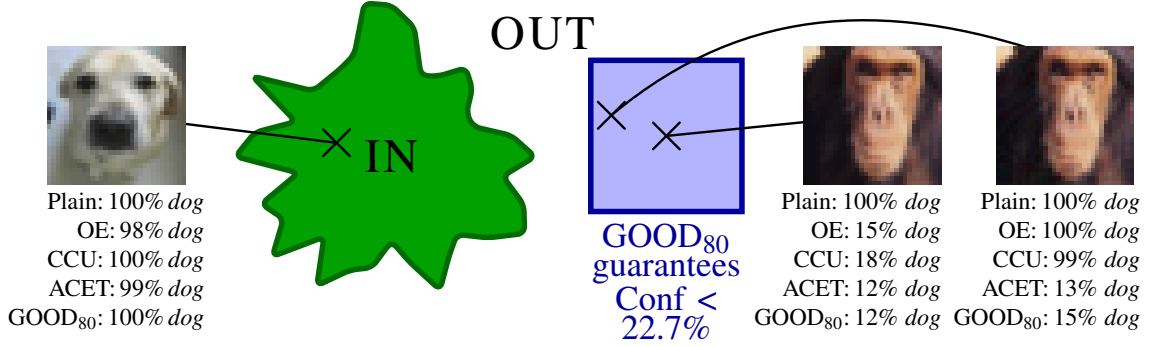


Figure 5.1: **Overconfident predictions on out-distribution inputs.** **Left:** On the in-distribution CIFAR-10 all methods have similar high confidence on the image of a *dog*. **Mid-dle:** For the out-distribution image of a *chimpanzee* from CIFAR-100 the plain model is overconfident while an out-distribution aware method like Outlier Exposure (OE) (Hendrycks et al., 2019a) or CCU (Meinke & Hein, 2020) produces low confidence. **Right:** When maximizing the confidence inside the l_∞ -ball of radius 0.01 around the *chimpanzee* image (for the OE model), OE as well as CCU become overconfident (right image). ACET and our GOOD₈₀ perform well in having empirical low confidence, but only GOOD₈₀ guarantees that the confidence in the l_∞ -ball of radius 0.01 around the *chimpanzee* image (middle image) is less than 22.7% for any class (note that 10% corresponds to maximal uncertainty as CIFAR-10 has 10 classes).

In this chapter, we aim to provide worst-case OOD guarantees not only for noise but also for images from related but different image classification tasks. For this purpose we use the techniques from interval bound propagation (IBP) (Gowal et al., 2018) to derive a provable upper bound on the maximal confidence of the classifier in an l_∞ -ball of radius ϵ around a given point. By minimizing this bound on the out-distribution using our training scheme GOOD (Guaranteed Out-Of-distribution Detection) we arrive at the first models which have guaranteed low confidence even on image classification tasks related to the original one; e.g., we get state-of-the-art results on separating letters from EMNIST from digits in MNIST even though the digit classifier has never seen any images of letters at training time. In particular, the guarantees for the training out-distribution generalize to other out-distribution datasets. In contrast to classifiers which have certified adversarial robustness on the in-distribution, GOOD has the desirable property to achieve provable guarantees for OOD detection with almost no loss in accuracy on the in-distribution task even on datasets like CIFAR-10. An illustration of the type of guarantee that we provide with GOOD is given in Figure 5.1.

5.2 Definitions, setup and baselines

Let $f : \mathbb{R}^d \rightarrow \mathbb{R}^K$ be a feedforward neural network (DNN) with a last linear layer where d is the input dimension and K the number of classes. In all experiments below we use the ReLU

activation function. The logits of $f(x)$ for $x \in \mathbb{R}^d$ are transformed via the softmax function into a probability distribution $p(x)$ over the classes with:

$$p_k(x) := \frac{e^{f_k(x)}}{\sum_{l=1}^K e^{f_l(x)}} \text{ for } k = 1, \dots, K. \quad (5.1)$$

By $\text{Conf}_f(x) = \max_{k=1, \dots, K} p_k(x)$ we define the confidence of the classifier f in the prediction $\arg \max_{k=1, \dots, K} p_k(x)$ at x .

As baselines and motivation for our provable approach we use the OOD detection methods Outlier Exposure (OE) (Hendrycks et al., 2019a) and Confidence Enhancing Data Augmentation (CEDA) (Hein et al., 2019), which use as objective for training

$$\frac{1}{N} \sum_{i=1}^N \mathcal{L}_{\text{CE}}(x_i^{\text{IN}}, y_i^{\text{IN}}) + \frac{\kappa}{M} \sum_{j=1}^M \mathcal{L}_{\text{OUT}}(x_j^{\text{OUT}}), \quad (5.2)$$

where $\{(x_i^{\text{IN}}, y_i^{\text{IN}}) \mid 1 \leq i \leq N\}$ is the in-distribution training set, $\{x_j^{\text{OUT}} \mid 1 \leq j \leq M\}$ the out-distribution training set, and $\mathcal{L}_{\text{CE}}$ the cross-entropy loss. The hyper-parameter κ determines the relative magnitude of the two loss terms and is usually chosen to be one (Hendrycks et al., 2019a; Meinke & Hein, 2020; Hein et al., 2019). OE and CEDA differ in the choice of the loss $\mathcal{L}_{\text{OUT}}$ for the out-distribution where OE uses the cross-entropy loss between $p(x_j^{\text{OUT}})$ and the uniform distribution and CEDA uses $\log \text{Conf}_f(x_j^{\text{OUT}})$. Note that both the CEDA and OE loss attain their global minimum when $p(x)$ is the uniform distribution. Their difference is typically minor in practice. An important question is the choice of the out-distribution. For general image classification, it makes sense to use an out-distribution which encompasses basically any possible image one could ever see at test time and thus the set of all natural images is a good out-distribution; following (Hendrycks et al., 2019a) we use the 80 Million Tiny Images dataset (Torralba et al., 2008) as a proxy for that.

While OE and CEDA yield state-of-the-art OOD detection performance for image classification tasks when used together with the 80M Tiny Images dataset as out-distribution, they are, similarly to normal classifiers, vulnerable to adversarial manipulation of the out-distribution images where the attack is trying to maximize the confidence in this scenario (Meinke & Hein, 2020). Thus (Hein et al., 2019) proposed Adversarial Confidence Enhanced Training (ACET) which replaces the CEDA loss with $\max_{\|\hat{x} - x_j^{\text{OUT}}\|_{\infty} \leq \epsilon} \log \text{Conf}_f(\hat{x})$ and can be seen as adversarial training on the out-distribution for an l_{∞} -threat model. However, as for to adversarial training on the in-distribution (Madry et al., 2018) this does not yield any guarantees for out-distribution detection. In the next section we discuss how to use interval-bound-propagation (IBP) to get guaranteed OOD detection performance in an l_{∞} -neighborhood of every out-distribution input.

5.2.1 AUC and Conservative AUC

As a measure for the separation of in- vs. out-distribution data we use the Area Under the Receiver Operating Characteristic curve (AUROC or AUC) using the confidence of the classifier as the feature. The AUC is equal to the empirical probability of a random in-sample to be assigned a higher confidence than a random out-sample, plus one half times the probability of the confidences being equal. Thus, the standard way (as e.g. implemented in scikit-learn (Pedregosa et al., 2011)) to calculate the AUC from given confidence values on sets of in- and out-distribution samples S_{in} and S_{out} is

$$\begin{aligned} \text{AUC}(f, S_{in}, S_{out}) = \frac{1}{|S_{in}||S_{out}|} & \left(\left| \{x_{in} \in S_{in}, x_{out} \in S_{out} \mid \text{Conf}_f(x_{in}) > \text{Conf}_f(x_{out})\} \right| \right. \\ & \left. + \frac{1}{2} \left| \{x_{in} \in S_{in}, x_{out} \in S_{out} \mid \text{Conf}_f(x_{in}) = \text{Conf}_f(x_{out})\} \right| \right), \end{aligned} \quad (5.3)$$

where for a set S , $|S|$ indicates the number of its elements. The half-weighted equality term gives this definition certain symmetry properties. However, it assigns a positive score to some completely uninformed functions f . For example, a constant uniform classifier with $p_k(x) = \frac{1}{K}$ receives an AUC value of 50%. Similarly, a classifier that assigns 100% confidence to most in-distribution inputs would have positive AUC and even GAUC statistics, even if it fails to have confidence below 100% on any OOD inputs. In order to regard only example pairs where the distributions are positively distinguished, we define the **Conservative AUC** (cAUC) by dropping the equality term:

$$\text{cAUC}(f, S_{in}, S_{out}) := \frac{1}{|S_{in}||S_{out}|} \left| \{x_{in} \in S_{in}, x_{out} \in S_{out} \mid \text{Conf}_f(x_{in}) > \text{Conf}_f(x_{out})\} \right|. \quad (5.4)$$

While in general $\text{cAUC}(f, S_{in}, S_{out}) \leq \text{AUC}(f, S_{in}, S_{out})$, the confidences of all models presented in the paper are differentiated enough so that for all shown numbers actually $\text{cAUC} = \text{AUC}$. However, we have experienced that one can have models where the confidences (uniform or one-hot predictions) cannot be distinguished due to limited numerical precision. In these cases the normal AUC definition would indicate a certain discrimination where it is actually impossible to discriminate the confidences.

5.3 GOOD: Provable guarantees for OOD detection

Our goal is to minimize the confidence of the classifier not only on the out-distribution images themselves but in a whole neighborhood around them. For this purpose, we first derive bounds on the maximal confidence on some l_∞ -ball around a given point. In certified adversarial robustness, IBP (Gowal et al., 2018) currently leads to the best guarantees for deterministic classifiers under the l_∞ -threat model. While other methods for deriving guarantees yield tighter

bounds (Wong & Kolter, 2018; Mirman et al., 2018), they are not easily scalable and, when optimized, the bounds given by IBP have been shown to be very tight (Gowal et al., 2018).

IBP. Interval bound propagation (Gowal et al., 2018) provides entrywise lower and upper bounds $\underline{z}_k$ resp. $\overline{z}_k$ for the output z_k of the k -th layer of a neural network given that the input x is varied in the l_∞ -ball of radius ε . Let $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ be a monotonically increasing activation function e.g. we use the ReLU function $\sigma(x) = \max\{0, x\}$ in the paper. We set $z_0 = x$ and $\underline{z}_0 = x - \varepsilon \cdot \mathbf{1}$ and $\overline{z}_0 = x + \varepsilon \cdot \mathbf{1}$ ($\mathbf{1}$ is the vector of all ones). If the k -th layer is linear (fully connected, convolutional, residual etc.) with weight matrix W_k , one gets upper and lower bounds of the next layer via forward propagation:

$$\begin{aligned}\overline{z}_k &= \sigma(\max(W_k, 0) \cdot \overline{z}_{k-1} + \min(W_k, 0) \cdot \underline{z}_{k-1} + b_k) \\ \underline{z}_k &= \sigma(\min(W_k, 0) \cdot \overline{z}_{k-1} + \max(W_k, 0) \cdot \underline{z}_{k-1} + b_k),\end{aligned}\quad (5.5)$$

where the min/max expressions are taken componentwise and the activation function σ is applied componentwise as well. Note that the derivation in (Gowal et al., 2018) is slightly different, but the bounds are the same. The forward propagation of the bounds is of similar nature as a standard forward pass and back-propagation w.r.t. the weights is relatively straightforward.

Upper bound on the confidence in terms of the logits. The log confidence of the model at x can be written as

$$\log \text{Conf}(x) = \max_{k=1, \dots, K} \log \frac{e^{f_k(x)}}{\sum_{l=1}^K e^{f_l(x)}} = \max_{k=1, \dots, K} -\log \sum_{l=1}^K e^{f_l(x) - f_k(x)}. \quad (5.6)$$

We assume that the last layer is affine: $f(x) = W_L \cdot z_{L-1}(x) + b_L$, where L is the number of layers of the network. We calculate the upper bounds of all K^2 logit differences as:

$$\begin{aligned}\max_{\|\hat{x}-x\|_\infty \leq \varepsilon} f_k(\hat{x}) - f_l(\hat{x}) &= \max_{\|\hat{x}-x\|_\infty \leq \varepsilon} W_{L,k} \cdot z_{L-1}(\hat{x}) + b_{L,k} - W_{L,l} \cdot z_{L-1}(\hat{x}) - b_{L,l} \\ &= \max_{\|\hat{x}-x\|_\infty \leq \varepsilon} (W_{L,k} - W_{L,l}) \cdot z_{L-1}(\hat{x}) + b_{L,k} - b_{L,l} \\ &\leq \max(W_{L,k} - W_{L,l}, 0) \cdot \overline{z_{L-1}(x)}^\varepsilon \\ &\quad + \min(W_{L,k} - W_{L,l}, 0) \cdot \underline{z_{L-1}(x)}^\varepsilon + b_{L,k} - b_{L,l} \\ &=: \overline{f_k(x) - f_l(x)}^\varepsilon,\end{aligned}\quad (5.7)$$

where $W_{L,k}$ denotes the k -th row of W_L and $b_{L,k}$ is the k -th component of b_L . Note that this upper bound of the logit difference can be negative and is zero for $l = k$. Using this upper bound

on the logit difference in Equation (5.6), we obtain an upper bound on the log confidence:

$$\max_{\|\hat{x}-x\|_\infty \leq \varepsilon} \log \text{Conf}(\hat{x}) \leq \max_{k=1,\dots,K} -\log \sum_{l=1}^K e^{-(\overline{f_k(x)} - f_l(x))^\varepsilon} \quad (5.8)$$

We use the bound in (5.8) to evaluate the guarantees on the confidences for given out-distribution datasets. However, minimizing it directly during training leads to numerical problems, especially at the beginning of training, when the upper bounds $\overline{f_k(x)} - f_l(x)^\varepsilon$ are very large for $l \neq k$, which makes training numerically infeasible. Instead, we rather upper bound the log confidence again by bounding the sum inside the negative log from below with K times its lowest term:

$$\begin{aligned} \max_{k=1,\dots,K} -\log \sum_{l=1}^K e^{-(\overline{f_k(x)} - f_l(x))^\varepsilon} &\leq \max_{k=1,\dots,K} -\log \left(K \cdot \min_{l=1,\dots,K} e^{-(\overline{f_k(x)} - f_l(x))^\varepsilon} \right) \\ &= \max_{k,l=1,\dots,K} \overline{f_k(x)} - f_l(x)^\varepsilon - \log K \end{aligned} \quad (5.9)$$

While this bound can considerably differ from the potentially tighter bound of Equation (5.8), it is often quite close as one term in the sum dominates the others. Moreover, both bounds have the same global minimum when all logits are equal over the l_∞ -ball. We omit the constant $\log K$ in the following as it does not matter for training.

The direct minimization of the upper bound in (5.9) is still difficult, in particular for more challenging in-distribution datasets like SVHN and CIFAR-10, as the bound $\max_{k,l=1,\dots,K} \overline{f_k(x)} - f_l(x)^\varepsilon$ can be several orders of magnitude larger than the in-distribution loss. Therefore, we use the logarithm of this quantity. However, we also want to have a more fine-grained optimization when the upper bound becomes small in the later stage of the training. Thus we define the Confidence Upper Bound loss $\mathcal{L}_{\text{CUB}}$ for an OOD input as

$$\mathcal{L}_{\text{CUB}}(x; \varepsilon) := \log \left(\frac{\left(\max_{k,l=1,\dots,K} \overline{f_k(x)} - f_l(x)^\varepsilon \right)^2}{2} + 1 \right). \quad (5.10)$$

Note that $\log(\frac{a^2}{2} + 1) \approx \frac{a^2}{2}$ for small a and thus we achieve the more fine-grained optimization with an l_2 -type of loss in the later stages of training which tries to get all upper bounds small. The overall objective of **fully applied Guaranteed OOD Detection training (GOOD₁₀₀)** is the minimization of

$$\frac{1}{N} \sum_{i=1}^N \mathcal{L}_{\text{CE}}(x_i^{\text{IN}}, y_i^{\text{IN}}) + \frac{\kappa}{M} \sum_{j=1}^M \mathcal{L}_{\text{CUB}}(x_j^{\text{OUT}}; \varepsilon), \quad (5.11)$$

where $\{(x_i^{\text{IN}}, y_i^{\text{IN}}) \mid 1 \leq i \leq N\}$ is the in-distribution training set and $\{x_j^{\text{OUT}} \mid 1 \leq j \leq M\}$ the

out-distribution. The hyper-parameter κ determines the relative magnitude of the two loss terms. During training we slowly increase this value and ε in order to further stabilize the training with GOOD.

Quantile-GOOD: trade-off between clean and guaranteed AUC. Training models by minimizing (5.11) means that the classifier gets severely punished if *any* training OOD input receives a high confidence upper bound. If OOD inputs exist to which the classifier already assigns high confidence without even considering the worst case, e.g. as these inputs share features with the in-distribution, it makes little sense to enforce low confidence guarantees. Later in the experiments we show that for difficult tasks like CIFAR-10 this can happen. In such cases the normal AUC for OOD detection gets worse as the high loss of the out-distribution part effectively leads to low confidence on a significant part of the in-distribution which is clearly undesirable.

Hence, for OOD inputs x which are not clearly distinguishable from the in-distribution, it is preferable to just have the “normal” loss $\mathcal{L}_{\text{CUB}}(x_j^{\text{OUT}}; 0)$ without considering the worst case. We realize this by enforcing the loss with the guaranteed upper bounds on the confidence just on some quantile of the easier OOD inputs, namely the ones with the lowest guaranteed out-distribution loss $\mathcal{L}_{\text{CUB}}(x; \varepsilon)$. We first order the OOD training set by the potential loss $\mathcal{L}_{\text{CUB}}(x; \varepsilon)$ of each sample in ascending order π , that is $\mathcal{L}_{\text{CUB}}(x_{\pi_1}^{\text{OUT}}) \leq \mathcal{L}_{\text{CUB}}(x_{\pi_2}^{\text{OUT}}) \leq \dots \leq \mathcal{L}_{\text{CUB}}(x_{\pi_M}^{\text{OUT}})$. We then apply the loss $\mathcal{L}_{\text{CUB}}(x; \varepsilon)$ to the lower quantile q of the points (the ones with the smallest loss $\mathcal{L}_{\text{CUB}}(x; \varepsilon)$) and take $\mathcal{L}_{\text{CUB}}(x; 0)$ for the remaining samples, which means no worst-case guarantees on the confidence are enforced:

$$\frac{1}{N} \sum_{i=1}^N \mathcal{L}_{\text{CE}}(x_i^{\text{IN}}, y_i^{\text{IN}}) + \frac{\kappa}{M} \sum_{j=1}^{\lfloor q \cdot M \rfloor} \mathcal{L}_{\text{CUB}}(x_{\pi_j}^{\text{OUT}}; \varepsilon) + \frac{\kappa}{M} \sum_{j=\lfloor q \cdot M \rfloor + 1}^M \mathcal{L}_{\text{CUB}}(x_{\pi_j}^{\text{OUT}}; 0). \quad (5.12)$$

During training we do this ordering on the part of each batch consisting of out-distribution images. On CIFAR-10, where the out-distribution dataset 80M Tiny Images is closer to the in-distribution, the quantile GOOD-loss allows us to choose the trade-off between clean and guaranteed AUC for OOD detection, similar to the trade-off between clean and robust accuracy in adversarial robustness.

5.4 Experiments

We provide experimental results for image recognition tasks with MNIST (LeCun et al., 1998), SVHN (Netzer et al., 2011) and CIFAR-10 (Krizhevsky & Hinton, 2009) as in-distribution datasets. We first discuss the training details, hyperparameters and evaluation before we present the results of GOOD and competing methods. Code is available under <https://gitlab.com/Bitterwolf/GOOD>.

5.4.1 Model architectures and data augmentation

For all experiments, we use deep convolutional neural networks consisting of convolutional, affine and ReLU layers. For MNIST, we use the large architecture from (Gowal et al., 2018), and for SVHN and CIFAR-10 a similar but deeper and wider model. The layer structure is laid out in Table 5.1. Data augmentation is applied to both in- and out-distribution images during training. For MNIST we use random crops to size 28×28 with padding 4 and for SVHN and CIFAR-10 random crops with padding 4 as well as the quite aggressive augmentation AutoAugment (Cubuk et al., 2019). Additionally, we apply random horizontal flips for CIFAR-10.

Table 5.1: Model architectures used for MNIST (L), SVHN (XL) and CIFAR-10 (XL) experiments. Each convolutional and non-final affine layer is followed by a ReLU activation. All convolutions use a kernel size of 3, a padding of 1, and stride of 1, except for the third convolution which has stride=2.

L	XL
CONV2D(64)	CONV2D(128)
CONV2D(64)	CONV2D(128)
CONV2D(128) _{s=2}	CONV2D(256) _{s=2}
CONV2D(128)	CONV2D(256)
CONV2D(128)	CONV2D(256)
LINEAR(512)	LINEAR(512)
LINEAR(10)	LINEAR(512)
	LINEAR(10)

5.4.2 GOOD training procedure

As it is the case with IBP training (Gowal et al., 2018) for certified adversarial robustness, we have observed that the inclusion of IBP bounds can make the training unstable or cause it to fail completely. This can happen for our GOOD training despite the logarithmic damping in the $\mathcal{L}_{\text{CUB}}$ loss in (5.10). Thus, in order to further stabilize the training similar to (Gowal et al., 2018), we use linear ramp up schedules for ε and κ . As radii for the l_∞ -perturbation model on the out-distribution we use $\varepsilon = 0.3$ for MNIST, $\varepsilon = 0.03$ for SVHN and $\varepsilon = 0.01$ for CIFAR-10 (note that $0.01 > \frac{2}{255} \approx 0.0078$). The chosen $\varepsilon = 0.01$ for CIFAR-10 is so small that the changes are hardly visible (see Figure 5.1). As parameter κ for the trade-off between cross-entropy loss and the GOOD regularizer in (5.11) and (5.12), we set $\kappa = 0.3$ for MNIST and $\kappa = 1$ for SVHN and CIFAR-10.

For the MNIST experiments, we use as optimizer SGD with 0.9 Nesterov momentum, with an initial learning rate of $\frac{0.005}{128}$ that is divided by 5 after 50, 100, 200, 300 and 350 epochs, with a total number of 420 training epochs. For the GOOD, CEDA and OE runs, the first two epochs only use in-distribution $\mathcal{L}_{CE}$; over the next 100 epochs, the value of κ is ramped up linearly from zero to its final value of 0.3 for GOOD/OE and 1.0 for CEDA, where it stays for the remaining 318 epochs. The ε value in the $\mathcal{L}_{CUB}$ loss for GOOD is also increased linearly, starting at epoch 10 and reaching its final value of 0.3 on epoch 130. CCU is trained using the publicly available code from (Meinke & Hein, 2020), where we modify the architecture, learning rate schedule and data augmentation to be the same as OE. The initial learning rate for the Gaussian mixture models is $1e - 5/\text{batchsize}$ and gets dropped at the same epochs as the neural network learning rate. Our more aggressive data augmentation implies that our underlying Mahalanobis metric is not the same as they used in (Meinke & Hein, 2020). The ACET model for MNIST is warmed up with two epochs on the in-distribution only, then four with $\kappa = 1.0$ and $\varepsilon = 0$, and the full ACET loss with $\kappa = 1.0$ and $\varepsilon = 0.3$ for the remaining epochs. The reason why we chose a smaller κ of 0.3 for the MNIST GOOD runs is that considering the large ε for which guarantees are enforced, training with higher κ values makes training unstable without improving any validation results.

For the SVHN and CIFAR-10 baseline models, we used the ADAM optimizer (Kingma & Ba, 2015) with initial learning rate $\frac{0.01}{128}$ for SVHN and $\frac{0.1}{128}$ for CIFAR-10 that was divided by 5 after 30 and 100 epochs, with a total number of 420 training epochs. For OE, κ is increased linearly from zero to one between epochs 60 and 360. The same holds for CCU which again uses the same hyperparameters as OE. Again, ACET is warmed up with two in-distribution-only and four OE epochs. Then it is trained with $\kappa = 1.0$ and $\varepsilon = 0.03/0.01$ (SVHN/CIFAR-10), with a shorter training time of 100 epochs (the same number as used in (Hein et al., 2019)).

In line with the experiences reported in (Gowal et al., 2018) and (Zhang et al., 2020), for GOOD training on SVHN and CIFAR-10 longer training schedules with slower ramping up of the $\mathcal{L}_{CUB}$ loss are necessary, as adding the out-distribution loss defined in Equation (5.10) to the training objective at once will overwhelm the in-distribution cross-entropy loss and cause the model to collapse to uniform predictions for all inputs, without recovery. In order to reduce warm-up time, we use a pre-trained CEDA model for initialization and train for 900 epochs. The learning rate is 10^{-4} in the beginning and is divided by 5 after epochs 450, 750 and 850. Due to the pre-training, we begin training with a small κ and already start with non-zero ε after epoch 4. Then, ε is increased linearly to its final value of 0.03 for SVHN and 0.01 for CIFAR-10, which is reached at epoch 204. Simultaneously, κ is increased linearly with a virtual starting point at epoch -2 to its final value of 1.0 at epoch 298.

Due to the tendency of IBP based training towards instabilities, the selection of hyperparameters was based on finding settings where training is reliably stable while guaranteed bounds over meaningful ε -radii are possible.

In order to explore the potential trade-off between the separation of in- and out-distribution for clean and perturbed out-distribution inputs (clean AUCs vs guaranteed AUCs - see below), we train GOOD models for different quantiles $q \in [0, 1]$ in (5.12) which we denote as GOOD_q

in the following. Here, $Q = 100q$ is the percentage of out-distribution training samples for which we minimize the guaranteed upper bounds on the confidence of the neural network in the l_∞ -ball of radius ε around the out-distribution point during training. Note that GOOD_{100} corresponds to (5.11) where we minimize the guaranteed upper bound on the worst-case confidence for all out-distribution samples, whereas GOOD_0 can be seen as a variant of OE or CEDA. A training batch consists of 128 in- and 128 out-distribution samples. Examples of OOD training batches with the employed augmentation and their quantile splits for a GOOD_{60} model are shown in Table 5.2.

For the training out-distribution, we use 80 Million Tiny Images (80M) (Torralba et al., 2008), which is a large collection of natural images associated to nouns in wordnet (Fellbaum, 2012). All methods get the same out-distribution for training and we are *neither* training *nor* adapting hyperparameters for each OOD dataset separately as in some previous work. Since CIFAR-10 and CIFAR-100 are subsets of 80M, we follow (Hendrycks et al., 2019a) and filter them out. As can be seen in the example batches in Table 5.2, even this reduced dataset still contains images from CIFAR-10 classes, which explains why our quantile-based loss is essential for good performance on CIFAR-10. We take a subset of 50 million images as OOD training set. Since the size of the training set of the in-distribution datasets (MNIST: 60,000; SVHN: 73,257; CIFAR-10: 50000) is small compared to 50 million, typically an OOD image appears only once during training.

5.4.3 Evaluation setup

For each method, we compute the test accuracy on the in-distribution task, and for various out-distribution datasets (not seen during training) we report the area under the receiver operating characteristic curve (AUC) as a measure for the separation of in- from out-distribution samples based on the predicted confidences on the test sets. As OOD evaluation sets we use FashionMNIST (Xiao et al., 2017), the Letters of EMNIST (Cohen et al., 2017), grayscale CIFAR-10, and Uniform Noise for MNIST, and CIFAR-100 (Krizhevsky & Hinton, 2009), CIFAR-10/SVHN, LSUN Classroom (Yu et al., 2015), and Uniform Noise for SVHN/CIFAR-10.

We are particularly interested in the worst case OOD detection performance of all methods under the l_∞ -perturbation model for the out-distribution. For this purpose, we compute the **adversarial AUC (AAUC)** and the **guaranteed AUC (GAUC)**. These AUCs are based on the maximal confidence in the l_∞ -ball of radius ε around each out-distribution image.

For the adversarial AUC, we compute a lower bound on the maximal confidence in the l_∞ -ball by using Auto-PGD (Croce & Hein, 2020) for maximizing the confidence of the classifier inside the intersection of the l_∞ -ball and the image domain $[0, 1]^d$. Auto-PGD uses an automatic stepsize selection scheme and has been shown to outperform PGD. We use an adaptation to our setting with 500 steps and 5 restarts on 1000 points from each test set. Vanishing gradients pose a significant challenge, so we also perform a transfer attack on all models and on MNIST, we even use an additional attack. We report the per-sample worst-case across attacks. Note that attacking these models on different out-distributions poses somewhat dif-

ferent challenges than classical adversarial attacks. Around the in-distribution models with good prediction performance are unlikely to be completely flat (and thus have zero gradient) in the whole region defined by an l_∞ -threat model. On the out-distribution, however, it is quite possible that all neurons in some layer return negative pre-activations which causes all gradients to be zero. Therefore the choice of initialization together with several restarts matters a lot as otherwise non-robust OOD detection models can easily appear to be robust. Moreover, the transfer attacks were necessary for some methods as otherwise the true robustness would have been significantly overestimated. Indeed even though we invested quite some effort into adaptive attacks which are specific for our robust OOD detection scenario, it might still be that the AAUC of some methods is overestimated. This again shows how important it is to get provable guarantees.

For the guaranteed AUC, we compute an upper bound on the confidence in the intersection of the l_∞ -ball with the image domain $[0, 1]^d$ via IBP using (5.8) for the full test set. These worst case/guaranteed confidences for the out-distributions are then used for the AUC computation.

We compare a normally trained model (Plain), the state-of-the-art OOD detection method Outlier Exposure (OE) (Hendrycks et al., 2019a), as well as CEDA and ACET, both of which we adjusted compared to Chapter 4 by training on 80M instead of synthetic noise, since this natural surrogate data is available and yields much stronger results. As CEDA performs very similar to OE, we omit it in the figures for better readability. The ε -radii for the l_∞ -balls are the same for ACET and GOOD. So far the only method which could provide robustness guarantees for OOD detection is Certified Certain Uncertainty (CCU) with a data-dependent Mahalanobis-type l_2 threat model. We use their publicly available code to train a CCU model with our architecture and we evaluate their guarantees for our l_∞ threat model.

5.4.4 Experimental results

In Tables 5.3&5.4 we present the results on all datasets.

GOOD is provably better than OE/CEDA with regard to worst case OOD detection. We note that for almost all OOD datasets GOOD achieves non-trivial GAUCs. Thus the guarantees generalize from the training out-distribution 80M to the test OOD datasets. For the easier in-distributions MNIST and SVHN, which are more clearly separated from the out-distribution, the overall best results are achieved for GOOD₁₀₀. For CIFAR-10, the clean AUCs of GOOD₁₀₀ are low even when compared to plain training. Arguably the best trade-off for CIFAR-10 is achieved by GOOD₈₀. Note that the guaranteed AUC (GAUC) of these models is always better than the adversarial AUC (AAUC) of OE/CEDA (except for EMNIST). Thus it is fair to say that the worst-case OOD detection performance of GOOD is provably better than that of OE/CEDA. As expected, ACET yields good AAUCs but has no guarantees. It is notable that GOOD₁₀₀ has close to perfect guaranteed OOD detection performance for MNIST on CIFAR-10/uniform noise and for SVHN on **all** out-distribution datasets.

GOOD achieves certified OOD performance with almost no loss in accuracy. While there is a small drop in clean accuracy for MNIST, on SVHN, with 96.3% GOOD₁₀₀ has a better clean accuracy than all competing methods. On CIFAR-10, GOOD₈₀ achieves an

accuracy of 90.1% which is better than ACET and only slightly worse than CCU and OE. This is remarkable as we are not aware of any model with certified *adversarial robustness on the in-distribution* which gets even close to this range; e.g. IBP (Gowal et al., 2018) achieves an accuracy of 85.2% on SVHN with $\epsilon = 0.01$ (we have 96.3%), on CIFAR-10 with $\epsilon = \frac{2}{255}$ they get 71.2% (we have 90.1%). Previous certified methods had even worse clean accuracy. Since a significant loss in prediction performance is usually not acceptable, certified methods have not yet had much practical impact. Thus we think it is an encouraging and interesting observation that properties different from adversarial robustness like worst-case out-of-distribution detection can be certified without suffering much in accuracy. In particular, it is quite surprising that certified methods can be trained effectively with aggressive data augmentation like AutoAugment.

Trade-off between clean and guaranteed AUC via Quantile-GOOD. As discussed above, for the CIFAR-10 experiments, our training out-distribution contains images from in-distribution classes. This seems to be the reason why GOOD₁₀₀ suffers from a significant drop in clean AUC, as the only way to ensure small loss $\mathcal{L}_{CUB}$, if in- and out-distribution can partially not be distinguished, is to reduce also the confidence on the in-distribution. This conflict is resolved via GOOD₈₀ and GOOD₉₀ which both have better clean AUCs. It is an interesting open question if similar trade-offs can also be useful for certified adversarial robustness.

EMNIST: distinguishing letters from digits without ever having seen letters. GOOD₁₀₀ achieves an excellent AUC of 99.0% for the letters of EMNIST which is, up to our knowledge, state-of-the-art. Indeed, an AUC of 100% should not be expected as even for humans some letters like i and l are indistinguishable from digits. This result is quite remarkable as GOOD₁₀₀ has never seen letters during training. Moreover, as the AUC just distinguishes the separation of in- and out-distribution based on the confidence, in Figure 5.2 we show some samples from EMNIST together with their prediction/confidences for all models. GOOD₁₀₀ has a mean confidence of 98.4% on MNIST but only 27.1% on EMNIST in contrast to ACET with 75.0%, OE 87.9% and Plain 91.5%. This shows that while the AUC’s of ACET and OE are good for EMNIST, these methods are still highly overconfident on EMNIST. Only GOOD₁₀₀ produces meaningful higher confidences on EMNIST, when the letter has clear features of the corresponding digit. We see that GOOD₁₀₀ produces low confidences for most letters when they show no digit-specific features. Interestingly it even rejects some letters that could easily be mistaken for digits by humans (“o”). The mean confidence values of the same selection of MNIST models for each letter of the alphabet for EMNIST are plotted in Figure 5.3. We observe that the mean confidence often aligns with the intuitive likeness of a letter with some digit: GOOD₁₀₀ has the highest mean confidence on the letter inputs “i” and “l”, which in many cases do look like the digit “1”. Again, the confidence of GOOD₁₀₀ on the letter “o”, which even humans often cannot distinguish from a digit “0”, is generally low. On the other hand, “y” receives a surprisingly high confidence, compared to other letters, so we conclude that GOOD₁₀₀ uses different features than humans in order to achieve its impressive performance on EMNIST.

Plain	9: 63.6	8: 97.9	6: 87.2	0: 100.0	2: 60.0	4: 75.6	9: 100.0	4: 100.0	1: 100.0
OE	9: 79.7	0: 26.8	0: 78.8	0: 100.0	0: 69.4	4: 80.9	9: 100.0	4: 100.0	1: 100.0
CCU	4: 96.4	8: 31.8	6: 90.6	0: 100.0	2: 81.8	4: 86.8	9: 100.0	4: 100.0	1: 100.0
ACET	4: 79.6	0: 38.4	0: 61.6	0: 99.7	2: 55.7	4: 94.4	9: 100.0	4: 98.4	1: 99.9
GOOD ₁₀₀	8: 10.0	8: 11.4	0: 16.1	8: 10.0	8: 44.7	4: 24.1	9: 57.2	8: 10.0	1: 95.9
									
Plain	5: 99.2	6: 82.1	6: 100.0	4: 99.9	7: 99.8	0: 100.0	4: 89.7	8: 90.8	8: 97.4
OE	5: 98.7	6: 67.8	6: 100.0	4: 98.8	7: 99.5	0: 100.0	1: 65.9	8: 97.5	8: 96.2
CCU	5: 96.2	8: 52.7	6: 100.0	4: 89.0	7: 99.0	0: 99.9	1: 78.7	8: 80.2	8: 78.5
ACET	5: 79.4	6: 37.6	6: 99.7	4: 48.6	7: 58.8	0: 98.8	1: 48.0	8: 87.5	8: 69.0
GOOD ₁₀₀	5: 12.1	8: 10.0	6: 10.9	4: 21.2	8: 10.0	8: 10.0	1: 20.4	8: 28.1	8: 28.6
									
Plain	5: 100.0	4: 97.6	0: 90.0	4: 99.8	6: 74.8	4: 76.9	4: 99.8	2: 100.0	
OE	5: 100.0	4: 85.1	0: 79.5	4: 99.5	4: 68.0	4: 65.2	4: 100.0	2: 100.0	
CCU	5: 100.0	4: 98.8	0: 99.7	4: 92.7	4: 72.6	4: 66.0	4: 100.0	2: 99.7	
ACET	5: 99.9	4: 82.8	0: 87.1	4: 96.9	0: 60.8	6: 50.9	4: 99.8	2: 81.8	
GOOD ₁₀₀	5: 10.1	4: 50.2	0: 11.2	4: 12.1	8: 10.0	8: 10.0	4: 74.8	8: 10.0	
									

Figure 5.2: Random samples from the letters in the OOD dataset EMNIST. The predictions and confidences of all methods trained on MNIST are shown on top. GOOD₁₀₀ is the only method which is **not** overconfident (e.g. “H”) unless the letter is indistinguishable from a digit (“I”).

5.5 Chapter conclusions

In this chapter we have proposed GOOD, a novel training method to achieve guaranteed OOD detection in a worst-case setting. The certified lower bounds for GOOD’s robust OOD detection are better than the upper bounds on the robustness of Outlier Exposure, a state-of-the-art OOD detection method. We especially highlight the certified high robust performance for detecting EMNIST letters when MNIST is ID, which is a particularly challenging out-distribution detection task. Remarkably, the test accuracy of GOOD is comparable to the one of normal training of the same model architecture, which shows that certified methods have the potential to be useful in practice even for more complex tasks. In the next chapter, we will introduce with ProoD (Provable out-of-Distribution detector) a method that separates the provably robust and standard parts of GOOD in order to obtain an improved trade-off between GAUC and standard AUC and also inherits the strong standard test accuracy of the best available architecture and training setups. Since both methods have the same objectives, we will discuss other recent developments on certifiably adversarially robust OOD detection at the end of the following chapter.

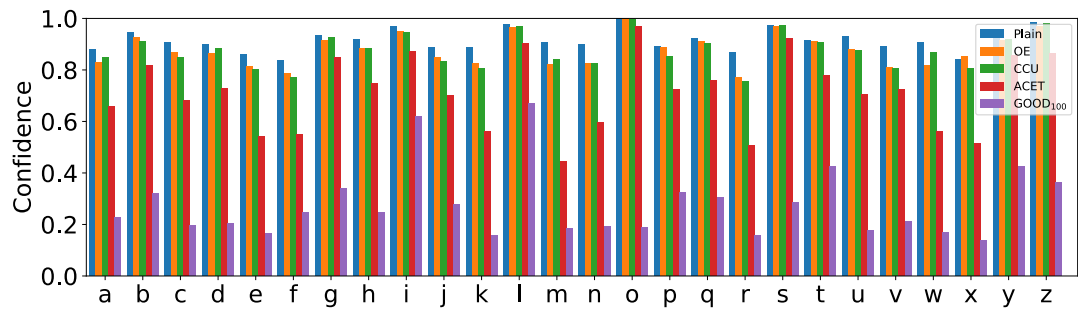


Figure 5.3: Mean confidence of different models across the classes of EMNIST-Letters. GOOD_{100} only has high mean confidence on letters that can easily be mistaken for digits.

Table 5.2: Quantile-GOOD training splits each OOD batch up into “harder” and “easier” parts, since trying to enforce low confidence guarantees on very close OOD results in very low confidences in general, even on the ID. Here we show exemplary batches of out-distribution 80M Tiny Images (after augmentation) towards the end of training of GOOD₆₀ models. **Top:** The 52 Images with highest confidence upper bound. On these, loss is based on standard output. **Bottom:** The remaining 76 Images with lowest confidence upper bound. Here, loss is based on upper bounds within the ε -ball. Even though the actual CIFAR images were filtered out, some images containing objects from CIFAR-classes are still present. For the CIFAR-10 model, such samples (among others) get sorted above the quantile. For MNIST, lower brightness images appear to be more difficult, while for SVHN images with fewer objects seem to be comparably hardest to distinguish from the house numbers of the in-distribution.

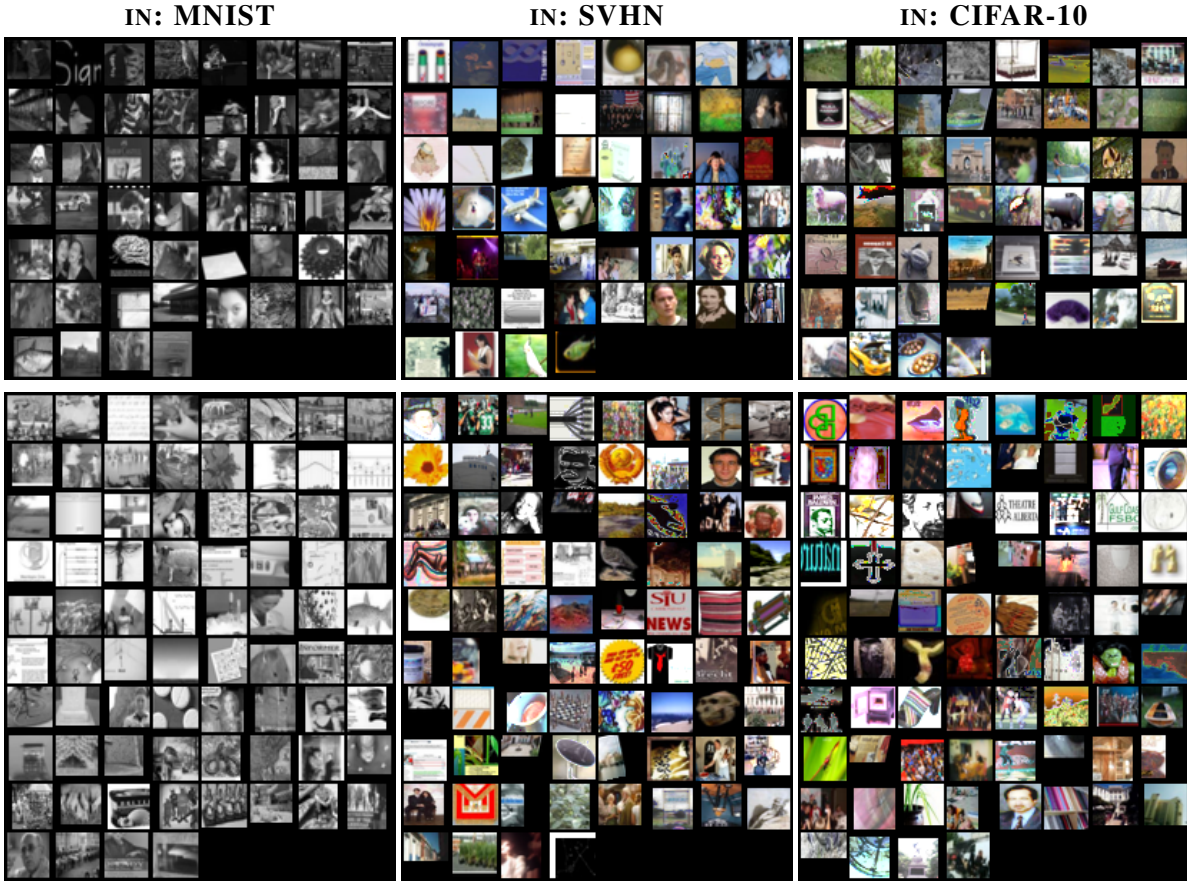


Table 5.3: **Accuracies as well as AUC, adversarial AUC (AAUC) and guaranteed AUC (GAUC) values** for the MNIST and SVHN in-distributions with respect to several unseen out-distributions. The GAUC of GOOD₁₀₀ is better than the corresponding AAUC of OE and CEDA on almost all OOD datasets (except EMNIST). Thus GOOD is provably better than OE and CEDA w.r.t. worst-case OOD detection. GOOD achieves this without significant loss in accuracy. Especially on SVHN, GOOD₁₀₀ has very good accuracy and almost perfect provably worst-case OOD detection performance.

IN: MNIST $\epsilon = 0.3$													
METHOD	ACC.	FASHIONMNIST			EMNIST LETTERS			CIFAR-10			UNIFORM NOISE		
		AUC	AAUC	GAUC	AUC	AAUC	GAUC	AUC	AAUC	GAUC	AUC	AAUC	GAUC
PLAIN	99.4	98.0	34.2	0.0	88.0	31.4	0.0	98.8	36.6	0.0	99.1	36.5	0.0
CEDA	99.4	99.9	82.1	0.0	92.6	52.8	0.0	100.0	95.1	0.0	100.0	100.0	0.0
OE	99.4	99.9	76.8	0.0	92.7	50.9	0.0	100.0	92.4	0.0	100.0	100.0	0.0
ACET	99.4	100.0	98.4	0.0	95.9	61.5	0.0	100.0	99.3	0.0	100.0	100.0	0.0
CCU	99.5	100.0	76.6	0.0	92.9	3.1	0.0	100.0	98.9	0.0	100.0	100.0	0.0
GOOD ₀	99.5	99.9	82.3	0.0	92.9	55.0	0.0	100.0	94.7	0.0	100.0	100.0	0.0
GOOD ₂₀	99.0	99.8	88.2	9.7	95.3	54.3	0.0	100.0	97.6	28.3	100.0	100.0	100.0
GOOD ₄₀	99.0	99.8	88.0	29.1	95.7	56.6	0.0	100.0	97.7	65.2	100.0	100.0	100.0
GOOD ₆₀	99.0	99.9	88.8	42.0	96.6	57.9	0.1	100.0	97.9	85.3	100.0	100.0	100.0
GOOD ₈₀	99.1	99.8	90.3	55.5	97.9	63.1	3.4	100.0	98.4	94.7	100.0	100.0	100.0
GOOD ₉₀	98.8	99.9	91.4	66.9	98.0	59.4	5.1	100.0	99.0	97.8	100.0	100.0	100.0
GOOD ₉₅	98.8	99.9	93.1	73.9	98.7	59.2	5.6	100.0	99.4	98.8	100.0	100.0	100.0
GOOD ₁₀₀	98.7	100.0	96.5	78.0	99.0	53.8	3.3	100.0	99.9	99.4	100.0	100.0	100.0

IN: SVHN $\epsilon = 0.03$													
METHOD	ACC.	CIFAR-100			CIFAR-10			LSUN CLASSROOM			UNIFORM NOISE		
		AUC	AAUC	GAUC	AUC	AAUC	GAUC	AUC	AAUC	GAUC	AUC	AAUC	GAUC
PLAIN	95.5	94.9	11.3	0.0	95.2	11.1	0.0	95.7	14.1	0.0	99.4	57.9	0.0
CEDA	95.3	99.9	63.9	0.0	99.9	68.7	0.0	99.9	80.7	0.0	99.9	99.3	0.0
OE	95.5	100.0	60.2	0.0	100.0	62.5	0.0	100.0	77.3	0.0	100.0	98.2	0.0
ACET	96.0	100.0	99.4	0.0	100.0	99.5	0.0	100.0	99.8	0.0	99.9	96.3	0.0
CCU	95.7	100.0	52.5	0.0	100.0	56.8	0.0	100.0	72.1	0.0	100.0	100.0	0.0
GOOD ₀	97.0	100.0	61.0	0.0	100.0	60.0	0.0	100.0	60.8	0.0	100.0	82.5	0.0
GOOD ₂₀	95.9	99.8	78.2	24.4	99.9	81.8	20.3	99.9	91.2	21.6	99.7	99.5	99.5
GOOD ₄₀	96.3	99.5	81.6	46.0	99.5	85.0	50.6	99.5	95.1	55.7	99.5	99.5	99.4
GOOD ₆₀	96.1	99.4	83.9	67.4	99.4	87.4	72.9	99.4	96.5	82.3	99.4	99.4	99.4
GOOD ₈₀	96.3	100.0	93.5	87.7	100.0	95.3	91.3	100.0	98.8	96.7	100.0	100.0	99.7
GOOD ₉₀	96.2	99.8	96.0	93.9	99.8	97.3	96.1	99.8	98.9	98.3	99.8	99.8	99.8
GOOD ₉₅	96.4	99.8	97.2	96.1	99.8	98.0	97.3	99.8	99.3	98.9	99.9	99.9	99.8
GOOD ₁₀₀	96.3	99.6	97.7	97.3	99.7	98.4	98.1	99.9	99.2	98.9	100.0	99.9	99.8

Table 5.4: **Accuracies as well as AUC, adversarial AUC (AAUC) and guaranteed AUC (GAUC) values** for the CIFAR-10 in-distribution with respect to several unseen out-distributions. The GAUC of GOOD₈₀ on CIFAR-10 is better than the corresponding AAUC of OE and CEDA on all OOD datasets. Thus GOOD is provably better than OE and CEDA w.r.t. worst-case OOD detection. GOOD achieves this without significant loss in accuracy.

IN: CIFAR-10 $\varepsilon = 0.01$													
METHOD	ACC.	CIFAR-100			SVHN			LSUN CLASSROOM			UNIFORM NOISE		
		AUC	AAUC	GAUC	AUC	AAUC	GAUC	AUC	AAUC	GAUC	AUC	AAUC	GAUC
PLAIN	90.1	84.3	13.0	0.0	87.7	10.6	0.0	88.9	13.6	0.0	90.8	56.4	0.0
CEDA	88.6	91.8	31.9	0.0	97.9	25.7	0.0	98.9	53.9	0.0	97.3	70.5	0.0
OE	90.7	92.4	11.0	0.0	97.6	3.7	0.0	98.9	20.0	0.0	98.7	75.7	0.0
ACET	89.3	90.7	74.5	0.0	96.6	88.0	0.0	98.3	91.2	0.0	99.7	98.9	0.0
CCU	91.6	93.0	23.3	0.0	97.1	14.8	0.0	99.3	38.2	0.0	100.0	100.0	0.0
GOOD ₀	89.8	92.9	22.5	0.0	97.0	12.8	0.0	98.3	48.4	0.0	96.3	95.6	0.0
GOOD ₂₀	88.5	90.3	32.4	11.8	95.9	28.3	15.8	98.2	48.2	3.4	99.4	97.6	87.5
GOOD ₄₀	89.5	89.6	38.2	24.8	95.4	38.0	24.9	96.0	62.0	27.4	92.1	89.9	89.8
GOOD ₆₀	90.2	88.6	42.6	34.9	95.6	44.4	39.0	97.0	67.6	49.1	91.8	91.3	91.2
GOOD ₈₀	90.1	85.6	48.2	42.3	94.0	41.4	38.0	93.3	66.9	55.2	95.8	95.4	95.3
GOOD ₉₀	90.2	81.7	51.5	49.6	91.4	48.7	46.9	90.2	63.5	57.7	89.3	87.7	87.7
GOOD ₉₅	90.4	80.3	52.0	50.8	90.2	44.4	43.3	88.3	62.6	60.3	96.6	95.9	95.8
GOOD ₁₀₀	90.1	70.0	54.7	54.2	75.5	58.9	56.9	75.2	61.5	61.0	99.5	99.2	99.0

Chapter 6

Separation of Certified and Standard OOD detection

This chapter is based on the paper “Provably Robust Detection of Out-of-distribution Data (almost) for free” (Meinke et al., 2022) which we presented at NeurIPS 2022. While he and I deliberated on potential improvements of the GOOD method as well as the implications of the possibility of separating OOD detection and ID classification, Alexander Meinke came up with the idea and potential benefits of certifying only the binary OOD detector. Alexander Meinke also developed the (semi)-joint training approach as well as the use of strictly negative weights for guarantees on asymptotic confidence and ran the experiments. Matthias Hein and I assisted with continual discussions and with the writing.

6.1 Improvement goals for certified worst case OOD detection

In the previous Chapter 5, we have developed the GOOD method, which produces a model whose confidence even on potentially adversarially perturbed OOD inputs is certifiably lower than on ID samples while simultaneously having only mildly impeded in-distribution accuracy compared to standard, non-robust training of the same architecture. From one perspective, this relatively strong performance can be regarded as an interesting result compared to certified adversarial classification robustness, whose robustness tends to come in hand with severely lowered clean accuracy. From another point of view, however, we see that in absolute terms the accuracy of GOOD models is still noticeably lower compared to what is achieved by standard classifier models with deeper architectures. For example on CIFAR10, a standard ResNet50 can reach 95% accuracy, while the models in Table 5.4 trained on the GOOD XL architecture have accuracies below 91%. In this chapter, we incorporate the insights on the potential separation of OOD detection and classification which we described in Chapter 3.

We again start by regarding OOD detection based on confidence, i.e. the softmax probability of the predicted class is used to decide whether to reject or accept the sample, because of its straightforward interpretation and because we saw in Chapter 3 that it performs no worse than other scores for OOD detection.

Table 6.1: ProofD combines desirable properties of existing (adversarially robust) OOD detection methods. It has high test accuracy and standard OOD detection performance (as (Hendrycks et al., 2019a)) and has worst-case guarantees if the out-distribution samples are adversarially perturbed in an l_∞ -neighborhood to maximize the confidence (see Section 6.4.2). Similar to CCU (Meinke & Hein, 2020) it avoids the problem of asymptotic overconfidence far away from the training data.

	OE	CCU	ACET/ATOM	GOOD	ProofD
High accuracy	✓	✓	✓		✓
High clean OOD detection performance	✓	✓	✓		✓
Adv. OOD l_∞ -robustness			(✓)	✓	✓
Adv. OOD l_∞ -certificates				✓	✓
Provably not asympt. overconfident		✓			✓

In this chapter, we still regard the robustness problem of standard OOD detection methods is that they are vulnerable to adversarial perturbations, i.e. small modifications of OOD inputs can lead to large confidence of the classifier on the manipulated samples (Nguyen et al., 2015; Hein et al., 2019; Schwag et al., 2019). Of course, an OOD input, which by definition is semantically far away from the in-distribution, should not be able to be moved into a region that is considered in-distribution by the detection model if the movement is imperceptibly small. On the other hand, as we discussed in Chapter 1, a slightly perturbed in-distribution input can still be considered in-distribution for some perturbations (e.g. if the perturbation resembles standard camera noise), but for other perturbations it might be highly atypical and therefore should arguably rather be seen as OOD. Furthermore, adversarial robustness on the in-distribution is known to come at the cost of clean accuracy (Tsipras et al., 2019) which hinders the adoption of such methods in practice. We aim to provide a method that does not harm the in-distribution performance in any way and thus, like with GOOD and other previous OOD-detection methods, we do *not* consider adversarially manipulated in-distribution samples and focus on ensuring that OOD samples remain OOD under adversarial attacks.

While different methods for adversarially robust OOD detection have been proposed (Hein et al., 2019; Schwag et al., 2019; Meinke & Hein, 2020; Chen et al., 2021; Bitterwolf et al., 2020) there is only a limited number of works on *provably* adversarially robust OOD detection (Meinke & Hein, 2020; Bitterwolf et al., 2020; Kopetzki et al., 2021; Berrada et al., 2021a). For the standard empirical evaluation of adversarial robustness, for each input one runs an array of different attacks that conform to the assumed threat model and records the output for the worst found perturbation. This means that there is no guarantee that a more malign perturbation does not exist, as only a lower bound on the adversarial robustness is established. Provable adversarial robustness on OOD data, yields a mathematically deduced upper bound on the worst-case confidence around each OOD sample. For our guaranteed upper bounds on the confidence of an OOD sample, it is certain that no applicable manipulation raises the confidence above the certified value.

In (Kopetzki et al., 2021) they apply randomized smoothing to obtain guarantees wrt. l_2 -perturbations for Dirchlet-based models (Malinin & Gales, 2018, 2019; Sensoy et al., 2018) which already show quite some gap in terms of AUC-ROC to SOTA OOD detection methods even without attacks. Interval bound propagation (IBP) (Gowal et al., 2018; Mirman et al., 2018; Zhang et al., 2020; Jovanović et al., 2022a) has been shown to be one of the most effective techniques in certified adversarial robustness on the in-distribution when applied during training. In GOOD we used IBP to compute upper bounds on the confidence in an l_∞ -neighborhood of the input and minimize these upper bounds on a training out-distribution. This yields classifiers with pointwise guarantees for adversarially robust OOD detection even for “close” out-distribution inputs which generalize to novel OOD test distributions. However, the employed architectures of the neural network are restricted to rather shallow networks as otherwise the bounds of IBP are loose. Thus, they obtain low classification accuracy which is far from the state-of-the-art, e.g. 91% on CIFAR10, and their approach does not scale to more complex tasks like ImageNet. In particular, despite its low accuracy the employed network architecture is quite large and has higher memory consumption than a ResNet50. The authors of (Berrada et al., 2021a) use SOTA verification techniques (Dathathri et al., 2020) and get guarantees for OOD detection wrt. l_∞ -perturbations for ACET models (Hein et al., 2019) that were not specifically trained to be verifiable but the guarantees obtained by training the models via IBP in (Bitterwolf et al., 2020) are significantly better.

A *different* type of guaranteed low confidence on OOD data pertains to the asymptotic behavior of a classifier. Since standard ReLU networks provably have increasing confidence in almost all directions far away from the training data (Hein et al., 2019), one has to modify the architecture in order to solve this problem. In CCU (Meinke & Hein, 2020) the authors append density estimators based on Gaussian mixture models for in- and out-distribution to the softmax layer. By also enforcing low confidence on a training out-distribution, they achieve similar OOD detection performance to (Hendrycks et al., 2019a) but can guarantee that the classifier shows decreasing confidence as one moves away from the training data. However, for close in-distribution inputs this approach yields no guarantee as the Gaussian mixture models are not powerful enough for complex image classification tasks. In (Kristiadi et al., 2020, 2021) similar asymptotic guarantees are derived for Bayesian neural networks but without any robustness guarantees.

We propose ProoD which merges a certified binary discriminator for in-versus out-distribution with a classifier for the in-distribution task in a principled fashion into a joint classifier. This combines the advantages of CCU (Meinke & Hein, 2020) and GOOD (Bitterwolf et al., 2020) without suffering from their downsides. In particular, ProoD simultaneously achieves the following:

- Guaranteed adversarially robust OOD detection via certified upper bounds on the confidence in l_∞ -balls around OOD samples.
- Additionally, it provably prevents the asymptotic overconfidence of deep neural networks.

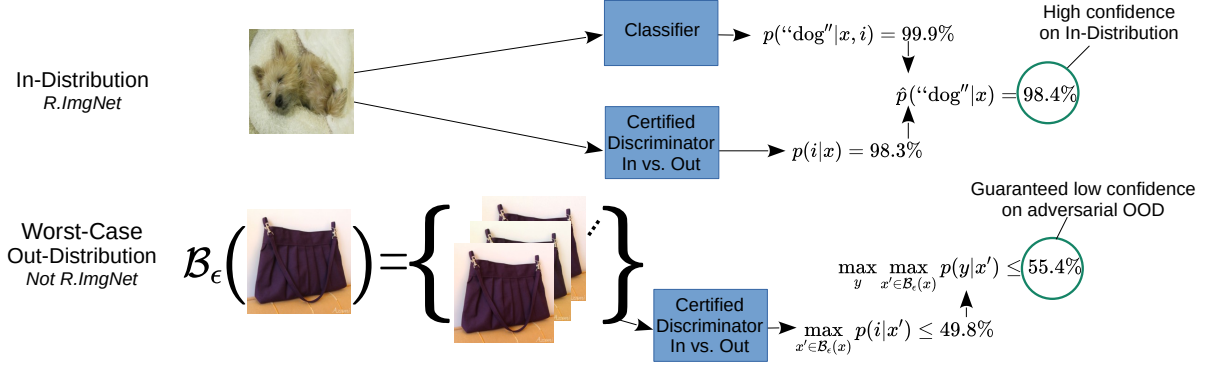


Figure 6.1: **ProoD's Architecture:** Combining the output of a classifier and a certified discriminator, see Eq. (6.2), we achieve high confidence on the in-distribution sample of a dog (R.ImgNet). The certified discriminator, see Eq. (6.4), yields an upper bound on the confidence in a ℓ_∞ -neighborhood of the shown OOD sample not belonging to any classes of R.ImgNet. ProoD achieves provable guarantees on adversarial OOD detection without loss in accuracy or clean OOD detection.

- It can be used with arbitrary architectures and has no loss in prediction performance and standard OOD detection performance.

Thus, we get provable guarantees for adversarially robust OOD detection, fix the asymptotic overconfidence (almost) for free as we have (almost) no loss in prediction and standard OOD detection performance. We qualitatively compare the properties of our model to prior approaches in Table 6.1. We provide code for all our experiments at <https://github.com/AlexMeinke/Provable-OOB-Detection>.

6.2 ProoD: The Provable out-of-Distribution detector

In the following we consider feedforward networks for classification, $f : \mathbb{R}^d \rightarrow \mathbb{R}^K$, with K classes defined with input $x^{(0)} = x$ and layers $l = 1, \dots, L-1$ as

$$x^{(l)} = \sigma^{(l)} \left(W^{(l)} x^{(l-1)} + b^{(l)} \right), \quad f(x) = W^{(L)} x^{(L-1)} + b^{(L)}, \quad (6.1)$$

where $W^{(l)}$ and $b^{(l)}$ are weights and biases and $\sigma^{(l)}$ is either the ReLU or leaky ReLU activation function of layer l . We refer to the output of f as the *logits* and get a probability distribution over the classes via $\hat{p}(y|x) = \frac{e^{f_y(x)}}{\sum_k^K e^{f_k(x)}}$ for $y = 1, \dots, K$. We define the confidence as $\text{Conf}(f(x)) = \max_{y=1, \dots, K} \hat{p}(y|x)$.

6.2.1 Joint Model for OOD Detection and Classification

In our joint model we assume that there exists an in- and out-distribution where the out-distribution samples are unrelated to the in-distribution task. Thus, we can formally write the conditional distribution on the input as

$$\hat{p}(y|x) = \hat{p}(y|x, i)\hat{p}(i|x) + \hat{p}(y|x, o)\hat{p}(o|x), \quad (6.2)$$

where $\hat{p}(i|x)$ is the conditional distribution that sample x belongs to the in-distribution and $\hat{p}(y|x, i)$ is the conditional distribution for the in-distribution. We assume that OOD samples are unrelated and thus maximally un-informative to the in-distribution task, i.e. we fix $\hat{p}(y|x, o) = \frac{1}{K}$, so that the classifier can be written as

$$\hat{p}(y|x) = \hat{p}(y|x, i)\hat{p}(i|x) + \frac{1}{K}(1 - \hat{p}(i|x)). \quad (6.3)$$

We train the binary classifier $\hat{p}(i|x)$ in a certified robust fashion wrt. an l_∞ -threat model so that even adversarially manipulated OOD samples are detected. In order to avoid confusion with the multi-class classifier, we will refer to $\hat{p}(i|x)$ as a binary discriminator. In an l_∞ -ball of radius ϵ around $x \in \mathbb{R}^d$ and for all y we get the upper bound on the confidence of the final classifier in Eq. (6.3):

$$\max_{\|x'-x\|_\infty \leq \epsilon} \hat{p}(y|x') \leq \max_{\|x'-x\|_\infty \leq \epsilon} \hat{p}(i|x') + \frac{1}{K}(1 - \hat{p}(i|x')) = \frac{K-1}{K} \max_{\|x'-x\|_\infty \leq \epsilon} \hat{p}(i|x') + \frac{1}{K}, \quad (6.4)$$

where we have used that $p(y|x, i) \leq 1 \forall x, y$, so we can defer the certification “work” to the binary discriminator. Using a particular constraint on the weights of the binary discriminator, we get similar asymptotic properties as in (Meinke & Hein, 2020) but additionally get certified adversarial robustness for close out-distribution samples as in (Bitterwolf et al., 2020). In contrast to (Bitterwolf et al., 2020), this comes without loss in test accuracy or non-adversarial OOD detection performance since in our model the neural network used for the in-distribution classification task $\hat{p}(y|x, i)$ is independent of the binary discriminator. Thus, we have the advantage that the classifier can use arbitrary deep neural networks and is not constrained to certifiable networks. We call our approach **Provable out-of-Distribution detector (ProoD)** and visualize its components in Figure 6.1. The intuitive idea of why ProoD can achieve adversarially robust OOD detection without loss in clean OOD detection can be explained with the behavior of the predicted probability distribution provided in Equation (6.3).

- **For clean OOD:** the classifier $\hat{p}(y|x, i)$ (trained similar to Outlier Exposure) already enforces low confidence on out-of-distribution points and thus irrespective of the values $\hat{p}(i|x)$, the resulting output of $\hat{p}(y|x)$ will be close to uniform as well and thus ProoD performs similar to Outlier Exposure.
- **For adversarial OOD:** the classifier confidence $\max_y \hat{p}(y|x, i)$ is potentially corrupted

but now the binary discriminator $\hat{p}(i|x)$ kicks in and ensures that the resulting prediction $\hat{p}(y|x)$ is close to uniform.

This explains why the combination of certified discriminator and classifier works much better than the individual parts and the use of this “redundancy” is the key idea of ProoD.

Certifiably Robust Binary Discrimination of In- versus Out-Distribution The first goal is to get a certifiably adversarially robust OOD detector $\hat{p}(i|x)$. We train this binary discriminator independently of the overall classifier as the training schedules for certified robustness are incompatible with the standard training schedules of normal classifiers. For this binary classification problem we use a logistic model $\hat{p}(i|x) = \frac{1}{1+e^{-g(x)}}$, where $g: \mathbb{R}^d \rightarrow \mathbb{R}$ are logits of a neural network (we denote the weights and biases of g by W_g and b_g in order to differentiate it from the classifier f introduced in the next paragraph). Let $(x_r, y_r)_{r=1}^N$ be our in-distribution training data (we use the class encoding +1 for the in-distribution and -1 for the out-distribution) and $(z_s)_{s=1}^M$ be our training out-distribution data. Then the optimization problem associated to the binary classification problem becomes:

$$\min_{W_g^{(L_g)} < 0} \frac{1}{N} \sum_{r=1}^N \log(1 + e^{-g(x_r)}) + \frac{1}{M} \sum_{s=1}^M \log(1 + e^{\bar{g}(z_s)}), \quad (6.5)$$

where we minimize over the parameters of the neural network g under the constraint that the weights of the output layer $W_g^{(L_g)}$ are componentwise negative and $\bar{g}(z) \geq \max_{u \in B_p(z, \varepsilon)} g(u)$ is an upper bound on the output of g around OOD samples for a given l_p -threat model $B_p(z, \varepsilon) = \{u \in [0, 1]^d \mid \|u - z\|_p \leq \varepsilon\}$. As before, we always use an l_∞ -threat model. This upper bound could, in principle, be computed using any certification technique but we will use interval bound propagation (IBP) since it is simple, fast and has been shown to produce SOTA results (Gowal et al., 2018). Note that this is not standard adversarial training for a binary classification problem as here we have an asymmetric situation: we want to be (certifiably) robust to adversarial manipulation on the out-distribution data but *not* on the in-distribution and thus the upper bound is only used for out-distribution samples. The negativity of the output layer’s weights $W_g^{(L_g)}$ is enforced by using the parameterization $(W_g^{(L_g)})_j = -e^{h_j}$ componentwise and optimizing over h_j . In Section 6.3 we show how the negativity of $W_g^{(L_g)}$ allows us to control the asymptotic behavior of the joint classifier.

For the reader’s convenience we quickly present the upper $\bar{x}^{(l)}$ and lower $\underline{x}^{(l)}$ bounds on the output of layer l in a feedforward neural network produced by IBP:

$$\bar{x}^{(l)} = \sigma(W_+^{(l)} \bar{x}^{(l-1)} + W_-^{(l)} \underline{x}^{(l-1)} + b^{(l)}), \quad \underline{x}^{(l)} = \sigma(W_+^{(l)} \underline{x}^{(l-1)} + W_-^{(l)} \bar{x}^{(l-1)} + b^{(l)}), \quad (6.6)$$

where $W_+ = \max(0, W)$ and $W_- = \min(0, W)$ (min/max used componentwise). For an l_∞ -threat model one starts with the upper and lower bounds for the input layer $\bar{x}^{(0)} = x + \varepsilon$ and

$\underline{x}^{(0)} = x - \varepsilon$ and then iteratively computes the layerwise upper and lower bounds $\bar{x}^{(l)}, \underline{x}^{(l)}$ which fulfill

$$\underline{x}^{(l)} \leq \min_{\|x' - x\|_\infty \leq \varepsilon} x^{(l)}(x') \leq \max_{\|x' - x\|_\infty \leq \varepsilon} x^{(l)}(x') \leq \bar{x}^{(l)}. \quad (6.7)$$

While in (Bitterwolf et al., 2020) they also used IBP to upper bound the confidence of the classifier this resulted in a bound that took into account all $\mathcal{O}(K^2)$ logit differences between all classes. In contrast, our loss in Eq. (6.5) is significantly simpler as we just have a binary classification problem and therefore only need a single bound. Thus, our approach easily scales to tasks with a large number of classes and training the binary discriminator with IBP turns out to be significantly more stable than the approach in (Bitterwolf et al., 2020).

(Semi)-Joint Training of the Final Classifier Given the certifiably robust model $\hat{p}(i|x)$ for the binary classification task between in- and out-distribution, we need to determine the final predictive distribution $\hat{p}(y|x)$ in Eq. (6.2). On top of the provable OOD performance that we get from Eq. (6.4), we also want to achieve SOTA performance on unperturbed OOD data. In principle we could independently train a model for the predictive in-distribution task $\hat{p}(y|x, i)$, e.g. using outlier exposure (OE) (Hendrycks et al., 2019a) or any other state-of-the-art OOD detection method and simply combine it with our $\hat{p}(i|x)$. While this does lead to models with high OOD performance that also have guarantees, it completely ignores the interaction between $\hat{p}(i|x)$ and $\hat{p}(y|x, i)$ during training. Instead we propose to train $\hat{p}(y|x, i)$ by optimizing our final predictive distribution $\hat{p}(y|x)$. Note that in order to retain the guarantees of $\hat{p}(i|x)$ we only train the parameters of the neural network $f: \mathbb{R}^d \rightarrow \mathbb{R}^K$ and need to keep $\hat{p}(i|x)$ resp. g fixed. Because g stays fixed we call this semi-joint training. We use OE (Hendrycks et al., 2019a) for training $\hat{p}(y|x)$ with the cross-entropy loss and use the softmax-function in order to obtain the predictive distribution $\hat{p}_f(y|x, i) = \frac{e^{f_y(x)}}{\sum_k e^{f_k(x)}}$ from f :

$$\begin{aligned} & \min_f -\frac{1}{N} \sum_{r=1}^N \log(\hat{p}(y_r|x_r)) - \frac{1}{M} \sum_{s=1}^M \frac{1}{K} \sum_{l=1}^K \log(\hat{p}(l|z_s)) \\ &= \min_f -\frac{1}{N} \sum_{r=1}^N \log\left(\hat{p}_f(y_r|x_r, i) \hat{p}(i|x_r) + \frac{1}{K} (1 - \hat{p}(i|x_r))\right) \\ & \quad - \frac{1}{M} \sum_{s=1}^M \frac{1}{K} \sum_{l=1}^K \log\left(\hat{p}_f(l|z_s, i) \hat{p}(i|z_s) + \frac{1}{K} (1 - \hat{p}(i|z_s))\right), \end{aligned} \quad (6.8)$$

where the first term is the standard cross-entropy loss on the in-distribution but now for our joint model for $\hat{p}(y|x)$ and the second term enforces uniform confidence on out-distribution samples.

The loss in Eq. (6.5) implicitly weighs the in-distribution and worst-case out-distribution equally, which amounts to the assumption $p(i) = \frac{1}{2} = p(o)$. This highly conservative choice simplifies training the binary discriminator but may not reflect the expected frequency of OOD samples at test time and in effect means that $\hat{p}(i|x)$ tends to be quite low. This typically yields

good guaranteed AUCs but can have a negative impact on the standard out-distribution performance. In order to better explore the trade-off of guaranteed and standard OOD detection, we repeat the above semi-joint training with different shifts of the offset parameter in the output layer

$$b' = b_g^{(L_g)} + \Delta, \quad (6.9)$$

where $\Delta \geq 0$ leads to increasing $\hat{p}(i|x)$. This shift has a direct interpretation in terms of the probabilities $p(i)$ and $p(o)$. Under the assumption that our binary discriminator g is perfect, that is

$$p(i|x) = \frac{p(x|i)p(i)}{p(x|i)p(i) + p(x|o)p(o)} = \frac{1}{1 + e^{-g(x)}}, \quad (6.10)$$

then it holds that $e^{g(x)} = \frac{p(x|i)p(i)}{p(x|o)p(o)}$. A change of the prior probabilities $\tilde{p}(i)$ and $\tilde{p}(o)$ without changing $p(x|i)$ and $p(x|o)$ then corresponds to a novel classifier

$$e^{\tilde{g}(x)} = \frac{p(x|i)\tilde{p}(i)}{p(x|o)\tilde{p}(o)} = \frac{p(x|i)p(i)}{p(x|o)p(o)} \frac{p(o)\tilde{p}(i)}{p(i)\tilde{p}(o)} = e^{g(x)} e^{\Delta} \quad (6.11)$$

with $\Delta = \log \left(\frac{p(o)\tilde{p}(i)}{p(i)\tilde{p}(o)} \right)$. Note that $\tilde{p}(i) > p(i)$ corresponds to positive shifts. In practice, this parameter can be chosen based on the priors for a particular application. Since no such priors are available in our case we determine a suitable shift by evaluating on the training out-distribution (see Section 6.4.2). Note that we explicitly do not train the shift parameter since this way the guarantees would get lost as the classifier implicitly learns a large Δ in order to maximize the confidence on the in-distribution, thus converging to a normal outlier exposure-type classifier without any guarantees.

6.3 Guarantees on asymptotic confidence

In this section we show that our specific construction provably avoids the issue of asymptotic overconfidence that was pointed out in (Hein et al., 2019). Note that the resulting guarantee (as stated in Theorem 6) is different from and in addition to the robustness guarantees discussed in the previous section (see Eq. (6.4)). The previous section dealt with providing confidence upper bounds on neighborhoods around OOD samples whereas this section deals with ensuring that a classifier's confidence decreases asymptotically as one moves away from all training data.

We note that a ReLU neural network $f : \mathbb{R}^d \rightarrow \mathbb{R}^K$ as defined in Eq. (6.1) using ReLU or leaky ReLU as activation functions, potential max-or average pooling and skip connection yields a piece-wise affine function (Arora et al., 2018), i.e. there exists a finite set of polytopes $Q_r \subset \mathbb{R}^d$ with $r = 1, \dots, R$ such that $\cup_{r=1}^R Q_r = \mathbb{R}^d$ and f restricted to each of the polytopes is an affine function. Since there are only finitely many polytopes some of them have to extend to infinity and on these ones the neural network is essentially an affine classifier. This fact has been used in (Hein et al., 2019) to show that ReLU networks are almost always asymptoti-

cally overconfident in the sense that if one moves to infinity the confidence of the classifier approaches 1 (instead of converging to the desirable $1/K$ as in these regions the classifier has never seen any data). The following theorem now shows that, in contrast to standard ReLU networks, our proposed joint classifier gets provably less confident in its decisions as one moves away from the training data which is a desirable property of any reasonable classifier.

Theorem 6. MainTheorem Let $x \in \mathbb{R}^d$ with $x \neq 0$ and let $g : \mathbb{R}^d \rightarrow \mathbb{R}$ be the ReLU-network of the binary discriminator (with the last activation being a non-leaky ReLU). Denote by $\{Q_r\}_{r=1}^R$ the finite set of polytopes on which g is affine (which exists as of Theorem 4 from Chapter 4). Denote by Q_t the polytope such that $\beta x \in Q_t$ for all $\beta \geq \alpha$ and let $x^{(L-1)}(z) = Uz + d$ with $U \in \mathbb{R}^{n_{L-1} \times d}$ and $d \in \mathbb{R}^{n_{L-1}}$ be the output of the pre-logit layer of g for $z \in Q_t$. If $Ux \neq 0$, then $\lim_{\beta \rightarrow \infty} \hat{p}(y|\beta x) = \frac{1}{K}$.

Proof. We note that with a similar argument as in the derivation of (6.4) it holds

$$\hat{p}(y|\beta x) \leq \hat{p}(i|\beta x) + \frac{1}{K}(1 - \hat{p}(i|\beta x)) = \frac{K-1}{K}\hat{p}(i|\beta x) + \frac{1}{K}. \quad (6.12)$$

Using Theorem 4 from Chapter 4, we know that there exist a polytope Q_t that is a linear region such that $\beta x \in Q_t$ for all $\beta \geq \alpha$. Thus for all $\beta \geq \alpha$ it holds that $\beta x \in Q_t$ such that

$$\hat{p}(i|\beta x) = \frac{1}{1 + e^{-g(\beta x)}} = \frac{1}{1 + e^{\langle W_g^{(L_g)}, U\beta x + d \rangle + b_g^{(L_g)}}}.$$

As $x_i^{(L-1)}(x) \geq 0$ for all $x \in \mathbb{R}^d$ it has to hold $(\beta Ux + d)_i \geq 0$ for all $\beta \geq \alpha$ and $i = 1, \dots, n_{L-1}$. This implies that $(Ux)_i \geq 0$ for all $i = 1, \dots, n_{L-1}$ and since $Ux \neq 0$ there has to exist at least one component i^* such that $(Ux)_{i^*} > 0$. Moreover, $W_g^{(L_g)}$ has strictly negative components and thus for all $\beta \geq \alpha$ it holds

$$g(\beta x) = \langle W_g^{(L_g)}, U\beta x + d \rangle + b_g^{(L_g)} = \beta \langle W_g^{(L_g)}, Ux \rangle + \langle W_g^{(L_g)}, d \rangle + b_g^{(L_g)}.$$

As $\langle W_g^{(L_g)}, Ux \rangle < 0$ we get $\lim_{\beta \rightarrow \infty} g(x) = -\infty$ and thus

$$\lim_{\beta \rightarrow \infty} \hat{p}(i|\beta x) = 0.$$

Plugging this into (6.12) yields the result. $\square$

Note that the negativity condition on the weights $W_g^{(L_g)}$ of the output layer of the in-vs. out-distribution discriminator g is crucial for the proof. This may seem restrictive, but we did not encounter any negative influence of this constraint on test accuracy, guaranteed or standard OOD detection performance. Thus, the asymptotic guarantees come essentially for free.

Table 6.2: **Architecture:** The architectures that are used for the binary discriminators. Each convolutional layer is directly followed by a ReLU.

CIFAR	R.ImgNet
Conv2d(3, 128)	Conv2d(3, 128)
Conv2d(128, 256) _{s=2}	AvgPool(2)
Conv2d(256, 256)	Conv2d(128, 256) _{s=2}
AvgPool(2)	AvgPool(2)
FC(16384, 128)	Conv2d(256, 256)
FC(128, 1)	AvgPool(2)
	FC(50176, 128)
	FC(128, 1)

6.4 Experiments

6.4.1 Training of ProoD

We conduct experiments with CIFAR10, CIFAR100 (Krizhevsky & Hinton, 2009) and Restricted Imagenet (R.ImgNet) (Tsipras et al., 2019) as in-distribution datasets. The latter is the same dataset as we used in Chapter 3 and consists of ImageNet images (ILSVRC2012) (Deng et al., 2009; Russakovsky et al., 2015) belonging to 9 types of animals, and as the training out-distribution, we again use the remaining ImageNet training samples (NotR.ImgNet). For the CIFAR in-distributions, we use OpenImages as training OOD data.

Training the Binary Discriminator We train the binary discriminator between in- and out-distribution using the loss in Eq. (6.5) with the bounds over an l_∞ -ball of radius $\varepsilon = 0.01$ for the out-distribution, which is what we already used with the GOOD method in Chapter 5. We use relatively shallow CNNs with only 5 layers plus pooling layers, see Table 6.2. Similarly to (Zhang et al., 2020) and Chapter 5, we use long training schedules, running Adam for 1000 epochs, with an initial learning rate of $1e-4$ that we decrease by a factor of 5 on epochs 500, 750 and 850 and with a batch size of 128 from the in-distribution and 128 from the out-distribution (for R.ImgNet: 50 epochs with drops at 25, 35, 45, batch sizes 32). In order to avoid large losses, as with GOOD, we also use a simple ramp up schedule for the ε used in IBP and we downweight the out-distribution loss during the initial phase of training by a scalar κ . Both ε and κ are increased linearly from 0 to their final values (0.01 and 1, respectively) over the first 300 epochs (for R.ImgNet over the first 25 epochs). Compared to the training of GOOD, we found that training of the binary discriminator is very stable and even 100 epochs on CIFAR would be sufficient, but we found that longer training leads to slightly better results. Weight decay is set to $5 \cdot 10^{-4}$, but is disabled for the weights in the final layer. We augment all

training images with AutoAugment (Cubuk et al., 2019) for CIFAR and with simple 4 pixel crops and reflections for R.ImgNet. The strict negativity of the weights leads to a negative bias of g which can cause problems at an early stage if the $b_g^{(L_g)}$ is initialized at 0 and thus we choose 3 as initialization. ”

Semi-joint training For the classifier we use a ResNet18 architecture on CIFAR and a ResNet50 on R.ImgNet. Note that the architecture of our binary discriminator is over an order of magnitude smaller than that of GOOD, and thus the memory overhead for the binary discriminator is less than a third of that of the classifier. On CIFAR we train for 100 epochs using SGD with momentum of 0.9 and a learning rate of 0.1 that drops by a factor of 10 on epochs 50, 75 and 90 (on R.ImgNet 75 epochs with drops at 30 and 60). For all datasets we train using a batch size of 128 (plus 128 out-distribution samples in the case of OE). As discussed in Section 6.2.1, when training the binary discriminator one implicitly assumes that in- and (worst-case) out-distribution samples are equally likely. It seems very unlikely that one would be presented with such a large number of OOD samples in practice but as discussed in Section 6.2.1, we can adjust the weight of the losses after training the discriminator (but before training the classifier) by shifting the bias $b_g^{(L_g)}$ in the output layer of the binary discriminator. We train several ProoD models for binary shifts in $\{0, 1, 2, 3, 4, 5, 6\}$ and then evaluate the AUC and guaranteed AUC (see 6.4.2) on a subset of the training out-distribution OpenImages (resp. NotR.ImgNet). For all bias shifts we use the same fixed provably trained binary discriminator and only train the classifier part. As our goal is to have provable guarantees with minimal or no loss on the standard OOD detection task, among all solutions which have better AUC than outlier exposure (OE) (Hendrycks et al., 2019a) we choose the one with the highest guaranteed AUC on OpenImages (on CIFAR10/CIFAR100) resp. NotR.ImgNet (on R.ImgNet). If none of the solutions has better AUC than OE on the training out-distribution we take the one with the highest AUC. We show the trade-off curves in Figure 6.2.

6.4.2 Evaluation

Setup For OOD evaluation for CIFAR10/100 we use the test sets from CIFAR100/10, SVHN (Netzer et al., 2011), the classroom category of downsampled LSUN (Yu et al., 2015) (LSUN_CR) as well as smooth noise as in Chapter 4. For R.ImgNet we use Flowers (Nilsback & Zisserman, 2008), FGVC Aircraft (Maji et al., 2013), Stanford Cars (Krause et al., 2013) and smooth noise as test out-distributions. Since the computation of adversarial AUCs (next paragraph) requires computationally expensive adversarial attacks, we restrict the evaluation on the out-distribution to a fixed subset of 1000 images (300 in the case of LSUN_CR) for the CIFAR experiments and 400 for the R.ImgNet models. We still use the entire test set for the in-distribution.

Guaranteed and adversarial AUC We again use the confidence of the classifier as the score to discriminate between in- and out-distribution samples and evaluate the adversarial

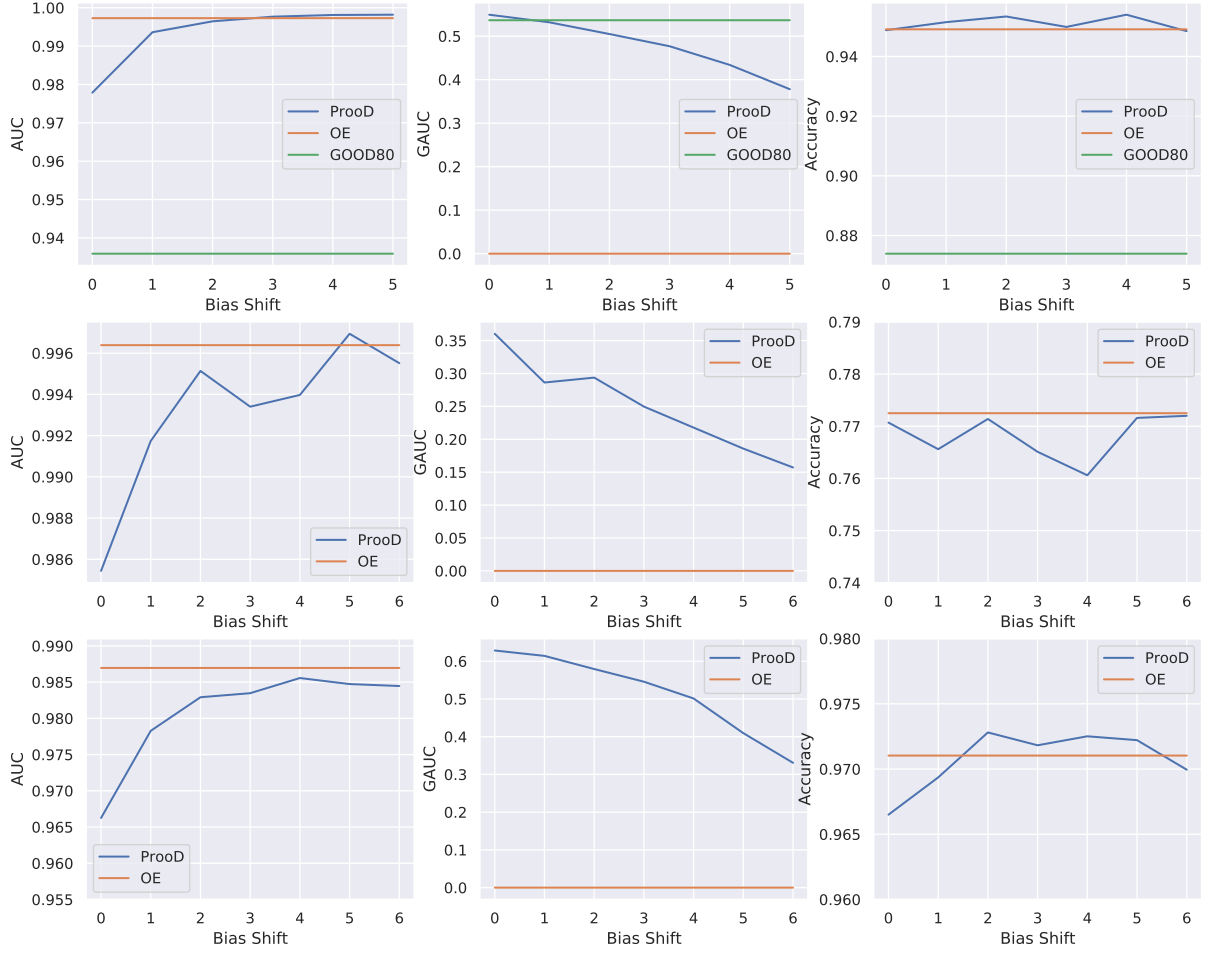


Figure 6.2: **Bias selection:** Using CIFAR10 (top), CIFAR100 (middle) and R.ImgNet (bottom) as the in-distribution and the test set of OpenImages (or NotR.ImgNet respectively) as OOD we plot the test accuracy, AUC and GAUC as a function of the bias shift Δ (see Eq. (6.9)).

AUC (AAUC) and the guaranteed AUC (GAUC) which we introduced in Chapter 5. Vanishing gradients (Papernot et al., 2017; Athalye et al., 2018a) are a significant challenge for the evaluation of AAUCs (Bitterwolf et al., 2020) even more than in the evaluation of adversarial robustness on the in-distribution as the models are trained to be “flat” on the out-distribution. Thus we use an ensemble of variants of projected gradient descent (PGD) (Madry et al., 2018) as well as the black-box SquareAttack (Andriushchenko et al., 2020a) with 5000 queries. We use APGD (Croce & Hein, 2020) (except on R.ImgNet, due to a memory leak) with 500 iterations and 5 random restarts. We also use a 200-step PGD attack with momentum of 0.9 and backtracking that starts with a step size of 0.1 which is halved every time a gradient step does not increase the confidence and gets multiplied by 1.1 otherwise. As stated above, the models are trained to be flat and thus the gradients can be exactly zero which renders gradient-based optimization impossible. Therefore it is important to use a variety of different starting points.

For PGD we start from: i) a decontrasted version of the image, i.e. the point that minimizes the l_∞ -distance to the grey image $0.5 \cdot \vec{1}$ within the threat model, ii) 3 uniformly drawn samples from the threat model, and iii) 3 versions of the original image perturbed by Gaussian noise with $\sigma = 10^{-4}$ and then clipped to the threat model. We always clip to $[0, 1]^d$ at each step of the attack. For all attacks and models we directly optimize the final score used for OOD detection.

Baselines We compare to a normally trained baseline (Plain) and outlier exposure (OE), both trained using the same architecture and hyperparameters as the classifier in ProoD. For comparisons to empirically robustified models, we include ACET, which in contrast to Chapter 4 we train with OpenImages as surrogate out-distribution instead of the synthetic smooth noise, as well as Chen et al. (2021)’s Adversarial Training with informative Outlier Mining (ATOM) method. For comparability with the literature and with previously published pre-trained models, we also include results with the 80M out-distribution in Table 6.4, where we also include the popular energy based OOD detector (Liu et al. (2020), see also Chapter 3) as a non-robust baseline. We note that with the attack procedure described above which is more adapted to the setting of OOD detectors with potentially flat loss landscapes than formerly used attack schemes, we found that the previously published AAUC values for ATOM and ACET in Chen et al. (2021) severely overestimate the true worst case AUC, as for example the AAUCs of the pre-trained CIFAR-100 models in Table 6.4 can in some cases be reduced to almost zero. This further exemplifies the benefit of the provable guarantees GOOD and ProoD can provide. For GOOD, we also retrain using OpenImages as training OOD dataset with the same code that we used in Chapter 5 where 80M Tiny Images was the surrogate out-distribution. While we had developed the training scheme for GOOD to be stable for MNIST, SVHN and CIFAR10, we tried to train models on CIFAR100 using the same code and same hyperparameters and schedules as we had used for CIFAR10. This only lead to models with accuracy below 25%, so we do not include these models in our evaluation. Since CCU was already shown to not provide benefits over OE on OOD data that is not very far from the in-distribution (e.g. uniform noise) (Meinke & Hein, 2020; Bitterwolf et al., 2020) we do not include it as baseline. We also evaluate the OOD-performance of the provable binary discriminator (ProoD-Disc) that we trained for ProoD. Note that this is not a classifier and is included only for reference. All results are in Table 6.3.

Results ProoD achieves non-trivial GAUCs on all datasets. As was also observed in (Bitterwolf et al., 2020), this shows that the IBP guarantees not only generalize to unseen samples but even to unseen distributions. In general, the gap between our GAUCs and AAUCs is extremely small. This shows that the seemingly simple IBP bounds can be remarkably tight, as has been observed in other works (Gowal et al., 2018; Jovanović et al., 2022a). It also shows that there would be very little benefit in applying stronger verification techniques like (Cheng et al., 2017; Katz et al., 2017; Dathathri et al., 2020) in ProoD. Similarly, it demonstrates the strengths of our attack as there provably does not exist an attack that could lower the AAUCs

Table 6.3: **Robust OOD detection:** For all models we report accuracy on the test set of the in-distribution and AUCs, guaranteed AUCs (GAUC), adversarial AUCs (AAUC) for different test out-distributions. The radius of the l_∞ -ball for the adversarial manipulations of the OOD data is $\varepsilon = 0.01$ for all datasets. The bias shift Δ that was used for ProoD is shown for each in-distribution. The AAUCs and GAUCs for ProoD tend to be very close, indicating remarkably tight certification bounds. Models with accuracy drop of $> 3\%$ relative to the model with highest accuracy are grayed out. Of the remaining models, we highlight the best OOD detection performance.

In: CIFAR10		CIFAR100			SVHN			LSUN_CR			Smooth		
	Acc.	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	95.01	90.0	0.0	0.7	93.8	0.0	0.3	93.1	0.0	0.5	98.0	0.0	0.7
OE	94.91	91.1	0.0	0.9	97.3	0.0	0.0	100.0	0.0	2.7	99.9	0.0	1.5
ATOM	93.63	78.3	0.0	21.7	94.4	0.0	24.1	79.8	0.0	20.1	99.5	0.0	73.2
ACET	93.43	86.0	0.0	4.0	99.3	0.0	4.6	89.2	0.0	3.7	99.9	0.0	40.2
GOOD ₈₀ *	87.39	76.7	47.1	57.1	90.8	43.4	76.8	97.4	70.6	93.6	96.2	72.9	89.9
GOOD ₁₀₀ *	86.96	67.8	48.1	49.7	62.6	34.9	36.3	84.9	74.6	75.6	87.0	76.1	78.1
ProoD-Disc	-	62.9	57.1	57.8	72.6	65.6	66.4	78.1	71.5	72.3	59.2	49.7	50.4
ProoD $\Delta=3$	94.99	89.8	46.1	46.8	98.3	53.3	54.1	100.0	58.3	59.7	99.9	38.2	38.8

In: CIFAR100		CIFAR10			SVHN			LSUN_CR			Smooth		
	Acc.	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	77.38	77.7	0.0	0.4	81.9	0.0	0.2	76.4	0.0	0.3	86.6	0.0	0.4
OE	77.25	77.4	0.0	0.2	92.3	0.0	0.0	100.0	0.0	0.7	99.5	0.0	0.5
ATOM	68.32	78.3	0.0	50.3	91.1	0.0	67.0	95.9	0.0	75.6	98.2	0.0	80.7
ACET	73.02	73.0	0.0	1.4	97.8	0.0	0.7	75.8	0.0	2.6	99.9	0.0	12.8
ProoD-Disc	-	56.1	52.1	52.3	61.0	58.2	58.4	70.4	66.9	67.1	29.6	26.4	26.5
ProoD $\Delta=5$	77.16	76.6	17.3	17.4	91.5	19.7	19.8	100.0	22.5	23.1	98.9	9.0	9.0

In: R.ImgNet		Flowers			FGVC			Cars			Smooth		
	Acc.	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	96.34	92.3	0.0	0.5	92.6	0.0	0.0	92.7	0.0	0.1	98.9	0.0	8.6
OE	97.10	96.9	0.0	0.2	99.7	0.0	0.4	99.9	0.0	1.8	98.0	0.0	1.9
ProoD-Disc	-	81.5	76.8	77.3	92.8	89.3	89.6	90.7	86.9	87.3	81.0	74.0	74.8
ProoD $\Delta=4$	97.25	96.9	57.5	58.0	99.8	67.4	67.9	99.9	65.7	66.2	98.6	52.7	53.5

*Uses different architecture of classifier, see “Baselines” in Section 6.4.2.

Table 6.4: **Training with 80M Tiny Images:** We repeat the evaluation from Table 6.3 for models that were trained using 80M Tiny Images as out-distribution instead of OpenImages. Plain is identical to before and is just repeated for the reader’s convenience. Note that the conclusions from OpenImages which we describe in the text still hold, which indicates that our method is robust to changes in the choice of training out-distribution. For ATOM and ACET we compare to pre-trained models from (Chen et al., 2021). Note that these models show almost no robustness on CIFAR100 - despite the far stronger claims in (Chen et al., 2021). Models with accuracy drop of $> 3\%$ relative to the model with highest accuracy are grayed out. Of the remaining models, we highlight the best OOD detection performance.

In: CIFAR10		CIFAR100			SVHN			LSUN_CR			Smooth		
	Acc.	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	95.01	90.0	0.0	0.6	93.8	0.0	0.1	93.1	0.0	0.5	98.2	0.0	0.6
OE	95.53	96.1	0.0	6.0	99.2	0.0	0.4	99.5	0.0	15.2	99.0	0.0	11.3
EB*	95.22	93.8	0.0	2.8	99.3	0.0	0.0	99.5	0.0	6.0	99.4	0.0	3.5
ATOM [†]	95.20	93.7	0.0	14.4	99.6	0.0	8.6	99.7	0.0	40.0	99.6	0.0	18.8
ACET [†]	91.48	91.2	0.0	80.5	95.3	0.0	87.6	98.9	0.0	95.0	99.9	0.0	98.3
GOOD ₈₀ *	90.13	87.2	42.5	63.9	94.2	37.5	67.4	93.3	55.2	83.6	95.3	57.3	88.5
GOOD ₁₀₀ *	90.14	70.7	54.5	55.0	74.9	56.3	56.6	75.2	61.0	61.6	81.4	66.6	67.5
ProoD-Disc	-	67.4	61.0	61.7	73.2	65.5	66.4	78.0	72.2	72.7	82.3	71.5	72.9
ProoD $\Delta=3$	95.47	96.0	41.9	43.9	99.5	48.8	49.4	99.6	47.6	53.1	99.7	55.8	57.0

In: CIFAR100		CIFAR10			SVHN			LSUN_CR			Smooth		
	Acc.	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	77.38	77.7	0.0	0.3	81.9	0.0	0.2	76.4	0.0	0.3	88.8	0.0	0.5
OE	77.28	83.9	0.0	0.8	92.8	0.0	0.1	97.4	0.0	4.6	97.6	0.0	0.9
EB*	75.70	77.4	0.0	0.8	96.5	0.0	0.0	96.7	0.0	5.9	98.9	0.0	4.3
ATOM [†]	75.06	64.3	0.0	0.2	93.6	0.0	0.2	97.5	0.0	9.3	98.5	0.0	15.0
ACET [†]	74.43	79.8	0.0	0.2	90.2	0.0	0.0	96.0	0.0	2.1	92.9	0.0	0.3
ProoD-Disc	-	53.8	50.3	50.4	73.1	69.8	69.9	68.1	63.8	64.0	67.2	63.8	63.9
ProoD $\Delta=1$	76.79	80.5	23.1	23.2	93.7	33.9	34.0	97.2	29.6	30.4	98.9	29.7	31.3

* Pre-trained WideResnet from (Liu et al., 2020).

[†]Densenet architecture, using models from (Chen et al., 2021) pre-trained with $\varepsilon = \frac{8}{255}$.

*CNN architecture using pre-trained models from (Bitterwolf et al., 2020).

on our ProoD model by more than 1.4% on any of the out-distributions. The bounds are also much tighter than for GOOD, which is likely due to the fact that for GOOD the confidence is much harder to optimize during an attack because it involves maximizing the confidence in an essentially random class.

For CIFAR10, on 3 out of 4 out-distributions ProoD’s GAUCs are higher than ATOM’s and ACET’s AAUCs, i.e. our model’s *provable* adversarial robustness exceeds the SOTA methods’ *empirical* adversarial robustness in these cases. Note that this is *not* due to our retraining, because the pre-trained models of Chen et al. (2021) perform even more poorly. On CIFAR100, ProoD’s guarantees are weaker and ATOM produces strong AAUCs. However, we observe that training both ACET and ATOM can produce inconsistent results, i.e. sometimes almost no robustness is achieved. For the successfully trained robust ATOM model on CIFAR100 we observe drastically reduced accuracy. Due to the difficulty in attacking these models, it is not unlikely that a more sophisticated attack could produce even lower AAUCs. Combined with the fact that both ACET and ATOM rely on expensive adversarial training procedures we argue that using ProoD is preferable in practice.

On CIFAR10, we see that ProoD’s GAUCs are comparable to, if slightly worse than the ones of both GOOD₈₀ and GOOD₁₀₀. Note that although the presented GOOD models are re-trained, the same observations hold true when comparing to the pre-trained models. However, we want to point out that ProoD achieves this while retaining both high accuracy and OOD performance, both of which are lacking for GOOD. It is also noteworthy that the GOOD models’ memory footprints are over twice as large as ProoD’s. Generally, for ProoD the accuracy is comparable to OE and the OOD performance is similar or marginally worse. Thus ProoD shows that it is possible to achieve certifiable adversarial robustness on the out-distribution while keeping very good prediction and OOD detection performance. Note that all methods struggle on separating CIFAR10 and CIFAR100 when using OpenImages as training OOD (as compared to 80M Tiny Images in Table 6.4).

To the best of our knowledge with R.ImgNet we provide the first worst case OOD guarantees on high-resolution images. The GAUCs are higher than on CIFAR, indicating that meaningful certificates on higher resolution are more achievable on this task than one might expect. FGVC and Cars may seem simple to separate from the animals in R.ImgNet but this cannot be said for Flowers which are difficult to provably distinguish from images of insects on flowers.

In summary, ProoD achieves our goal of maintaining high accuracy and clean OOD detection performance while providing provably adversarially robust OOD detection. In fact, out of all the methods that do not significantly impair the in-distribution accuracy, ProoD is the only method providing such guarantees as well while simultaneously having the highest empirical robustness. Also note that for applications where adversarial robustness on the in-distribution is desired despite the induced reduction in accuracy, one can combine our ProoD model with a robustly trained classifier.

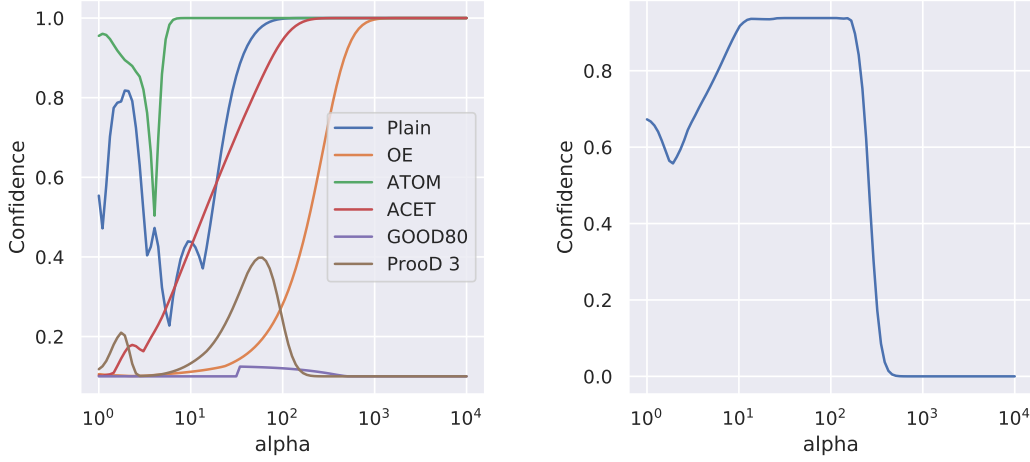


Figure 6.3: **Left, Asymptotic confidence:** We plot the mean confidence in the predicted in-distribution class for different models as one moves away from CIFAR100 samples along the trajectories $x + \alpha n$, where $n \in [-0.5, 0.5]^d$ and $\alpha \geq 0$. Only GOOD and ProoD converge to uniform confidence. **Right, Adversarial asymptotic confidence:** We try to find adversarial directions in which ProoD remains at a constant high confidence, as opposed to converging to low confidence. We plot the *maximum* of $\hat{p}(i|x)$ across 100 adversarially chosen directions as one moves further in these directions by factors of α . Note that $\hat{p}(i|x) \rightarrow 0$ implies $\hat{p}(y|x) \rightarrow \frac{1}{K}$.

6.5 Adversarial asymptotic overconfidence

As we found in Chapter 4, under mild conditions, we should expect to find asymptotic overconfidence in all ReLU networks and almost all directions. In order to empirically evaluate this, we take different models that were trained on CIFAR10 and evaluate their confidence on different CIFAR100 samples. For each sample x we track the confidence, $\max_k \hat{p}(k|x)$, along a trajectory in a uniform noise direction $x + \alpha n$, where $n \in [-0.5, 0.5]^d$ and $\alpha \geq 0$. The mean confidence across 100 such trajectories is shown on the left side of Figure 6.3. Even the models that produce low confidences on the original OOD sample asymptotically converge to maximal confidence far away. The only exceptions here are GOOD and ProoD and only ProoD can guarantee that the confidence cannot converge to 1.

However, even though the architecture provably prevents arbitrarily overconfident predictions and Theorem 6 ensures that most directions will indeed converge to uniform, it is, in principle, possible to find directions where the confidence $\hat{p}(i|x)$ remains constant if the condition $Ux \neq 0$ in Theorem 6 is not satisfied. We attempted to find such directions by running the following type of attack. We start from a random point $x \in [-0.5, 0.5]^d$ that we project onto a sphere of radius 100. We now run gradient descent (for 20000 steps), maximizing $g(x)$ while projecting onto the sphere at each step (unnormalized gradients with step size 0.1 for the first 10000 steps and 0.01 for the last 10000 steps). We then increase the radius to 1000 and run an additional 20000 steps with step size 0.1. We rescale the resulting direction vector down to an l_∞ -ball of norm 1 and compute the confidence $\hat{p}(i|x)$ as a function of the scaling

in the adversarial directions. We show the resulting scale-wise *maximum* over 100 adversarial directions in Figure 6.3. Note that even the worst-case over 100 adversarially found directions decays to 0 asymptotically, thus empirically confirming the practical utility of Theorem 6. Note that the value of $\hat{p}(i|x)$ converging to 0 implies that the confidence of the ProoD model $\hat{p}(y|x)$ converges to 10%.

In Figure 6.3 GOOD also stands out as having low confidence in all directions that we studied. This is because in all the asymptotic regions that we looked at, the pre-activations of the penultimate layer are all negative. If one moves outward and these pre-activations only get more negative in all directions far away from the data, the confidence does, in fact, remain low. Unfortunately, it also leads to gradients that are precisely zero, which is why the same attack can not be applied here. However, there is no guarantee that GOOD does not also get in some direction asymptotically overconfident.

6.6 Chapter conclusions

We have demonstrated how to combine a provably adversarially robust binary discriminator between in- and out-distribution with a standard classifier in order to simultaneously achieve high accuracy, high clean OOD detection performance as well as certified adversarially robust OOD detection. Thus, we have combined the best properties of previous work with only a small increase in total model size and only a single hyperparameter. This suggests that certifiable adversarial robustness on the out-distribution (as opposed to the in-distribution) is indeed possible without losing accuracy. We further showed how in our model simply enforcing negativity in the final weights of the discriminator fixes the problem of asymptotic overconfidence in ReLU classifiers. Training ProoD models is simple and stable and thus ProoD provides OOD guarantees that come (almost) for free.

Recent developments

Berrada et al. (2021b) derive probabilistic GAUC bounds for stochastic neural networks like Bayesian NNs Wicker et al. (2020) and MC Dropout networks (Gal & Ghahramani, 2016); these bounds are weaker than those of GOOD, but do not require the models to be specifically trained for provably robust OOD detection. Kotha et al. (2023) approximate the convex hull of the set of inputs that are accepted as ID and thus obtain provably robust OOD detection for data distributions with limited complexity in low dimensions. Kopetzki et al. (2021) apply randomized smoothing to uncertainty estimates in order to achieve probabilistically certified adversarially robust OOD detection.

Chapter 7

Conclusions and Outlook

Robustness of deep neural networks is an important, but arguably largely unsolved problem. In this dissertation, we regarded robustness against two types of issues that can arise when applying deep neural network classifiers after they have been trained: irate behaviour on out-of-distribution inputs and the vulnerability to adversarial perturbations. After illustrating and analyzing those problems, we eventually showed that it is possible to construct deep neural networks that are robust, at least to a degree, when encountering out-of-distribution inputs which have been subject to adversarial attacks.

We formulated explicit semantic requirements for an OOD datapoint such that a model's failure to detect it represents a clear weakness to OOD inputs. By meticulously examining the pre-existing datasets on which the research community had previously relied for evaluating OOD detection prowess of ImageNet models, we found in Chapter 2 (based on Bitterwolf et al. (2023)) that none of these datasets satisfies the requirement of its samples being OOD, while also being challenging to distinguish from ID data for standard OOD detectors. To cure this deficiency, we curated the NINCO (No ImageNet Class Objects) dataset, which serves as a challenging OOD dataset where every individual sample is scrutinized to be free of any objects that belong to one of the ImageNet classes. This allowed us to draw assured insights on the demeanour of prevailing OOD detectors and how their performance depends both on the choice of OOD detection scoring method and on the deep neural network to which that method is applied. The NINCO dataset also paves the way for future works to gain insights on the concrete proficiencies and shortcomings of OOD detectors, since it for the first time allows to regard individual inputs on which a detector makes the most confident errors, an analysis which with previous datasets would mostly bring up samples where the model correctly identifies ID objects in samples erroneously marked as OOD.

We went on in Chapter 3 (based on Bitterwolf et al. (2022)) to analyze the objective functions of neural network training methods that aim at obtaining classifier models with strong OOD detection capabilities, and compared them with each other as well as with mathematical combinations of the outputs of OOD unaware classifiers and binary ID vs. OOD discrimination models. For methods that incorporate surrogate OOD data during training, we showed that several methods indeed approximate equivalent objectives, but also found that training a single model — such that both OOD detection and ID classification share the same optimized parameters up until the penultimate network layer — yields far better results than modelling

both tasks completely separately. We also showed that the theoretical objective of ID density estimation models is equivalent to distinguishing ID samples from randomly drawn uniform noise samples, which might explain their inconsistent aptitude as OOD detectors. The connection between densities of ID and OOD data distributions and the construction of OOD detection methods can guide the design of new methods and also provide an intuition on the use cases in which certain approaches might be more or less suited than others.

Following those chapters on 'benign' OOD detection, we examined the interplay between OOD detection and adversarial robustness. We proposed and examined the threat model where a potential attacker tries to fool a detector model into falsely accepting OOD inputs as ID. The ACET models, which we introduced in Chapter 4 (based on Hein et al. (2019)) allow us to detect OOD samples even if they were adversarially perturbed towards appearing like they are in-distribution. To achieve this, they are trained to express low confidence on versions of OOD inputs with an adversarial perturbations that aims at provoking high confidence by the current model in each training step. We also highlighted and derived a theoretical explanation for the issue that when moving ever further away from the training data, the confidence of standard trained ReLU networks, which include standard convolutional neural networks, will converge towards 100% for almost all directions. We found that the ACET defense against adversarial increases of confidence on OOD data also solves this problem of asymptotic overconfidence.

Eventually, we devised the GOOD and ProoD methods in Chapters 5&6 (based on Bitterwolf et al. (2020) and Meinke et al. (2022)) which even provide certificates for adversarially robust detection with respect to the aforementioned threat model. Remarkably, their adversarial robustness against perturbed OOD samples does not come at a cost for the models' performance on in-distribution data, and the guarantees generalize to a range of distributions of OOD data that are remote from the surrogate examples that the model is shown during training. Training the GOOD models to achieve low confidence guarantees on OOD inputs while still being able to correctly classify ID samples with relatively higher confidence necessitated a certain amount of engineering. This included the introduction of loss quantiles which means that during training, we are content when on the hardest training samples, the model has low confidence but cannot provide guarantees, while it does provide them on the lion's share of the surrogate OOD training dataset. ProoD applies the insights from Chapter 3 towards certifiably adversarially robust OOD detectors, as training the robust ID vs. OOD detection problem and the ID classifier as individual parts (though semi-jointly), and for inference combining the former with the latter in order to achieve reasonable benign OOD detection. This makes training these models much smoother and allows us to obtain improved certification bounds simultaneously with strongly improved ID accuracy, which stems both from the less constraint classifier training and the possibility of employing vastly deeper network architectures for the classification task. Besides these certificates on OOD image test datasets, ProoD also introduces a last layer that models linear weights that are negative in each component, which does not negatively impact performance and guarantees convergence to low confidence when moving asymptotically far away from the training data, i.e. also provably solving the other issue against which ACET provided an empirical mitigation.

Still, there remain many crucial robustness issues, even when one only regards the inter-

section of the two aspects of adversarial robustness and OOD detection. For example, direct quantitative improvements like a higher clean, empirically robust and certifiedly robust OOD detection AUC (or FPR@95%TPR, or other measures) are obvious improvements. Narrowing down onto the perfect solution for a given scenario from the other side, by deriving theoretical and practical results on inputs types for which a correct solution cannot be defined as ground truth, or can at least not be learned under appropriate assumptions, could, if possible, lead to at least equally important results, indicating when further improvement efforts on the aforementioned metrics can be expected to be futile, and when the problem can be considered solved. With the absence of such upper bounds, it is an open question how and how far reliable OOD detection can be improved in the future. The complication of estimating what OOD detection performance can be hoped for in the best case is exacerbated by the relative vagueness of the scope of OOD inputs that might be relevant at inference time. In typical settings, they are difficult to formally enclose clearly and definitely, as no single out-distribution describes what unknown inputs can be encountered in any unforeseen situation. Thus, a notion of perfect OOD detection might not be meaningfully achievable. Even for a fixed out-distribution, represented by an OOD dataset that is fully separated from the in-distribution, it is not clear whether a model that has no information about this distribution can possibly be trained to accomplish perfect detection. On the other hand, humans appear to be just doing fine solving this task, which can be seen as evidence that it is in principle solvable when including certain assumptions that likely include modelling access to unrelated data and other circumstances that are relevant for the information that humans can be considered 'pre-trained' on.

Even in the case where the best possible detector can be derived, whether such an optimal solution would be adequate in a given application could still be open, and potentially warrant modifications to the original problem assumptions, like the collection of additional training data. Further straightforward room for improvement includes the extension of the adversarially robust radii with respect to the l_∞ threat models we investigated in this dissertation, as well as the consideration of other types of permitted attacker manipulations, which might be representative of certain relevant real-world attackers or incidental perturbations. With larger hardware scales becoming available come both the possibility and necessity to extend the proposed and analyzed methods to larger datasets with larger individual images. These pose several questions to established methods, including how they can be implemented and trained, whether they are still effective, and whether the choice of the best methods changes for larger scale datasets. These questions of transferability of methods are even starker for modalities different from image classification, the task on which both OOD detection and adversarial robustness have been predominantly studied until recently.

As we observed with the evaluations on the NINCO dataset, there are methods which significantly outperform the direct use of classifier confidence as a score to reject OOD samples. However, these methods are still far from perfect, as all of them still miss a significant fraction of OOD inputs, and many methods considered by the community to be effective OOD detectors even fail on simple OOD unit tests.

There are also several adjacent, but in their technical nature quite different, properties that can be included in the concept of robust OOD detection that would be highly desirable for

AI models used in safety critical or otherwise sensitive contexts. For example, one type of certified OOD detection that would be immensely relevant to many applications would be a guarantee that no OOD input from a certain set of reasonable distributions will be undetected. In practical settings where one has to deal with natural OOD samples that stem from different real-world occurrences or classes than the ID data, however, this goal seems to be extremely difficult to achieve. Even empirically detecting absolutely every OOD sample, without providing a theoretical guarantee, from a natural distribution is in many cases not achievable. This can be explained by the fact that the detection is at least as hard as solving a two class discrimination problem with some false negatives but no false positives being acceptable. Standard vision classifiers often fall short empirically of solving this base problem, and much less are able to guarantee perfect false positive rates. For OOD detection, this task gets markedly more complicated due to the unavailability of samples from the inference time OOD data during training. On the other hand, statistical performance guarantees involving provable approximate correctness or conformal prediction guarantees can provide some generalization estimates to new data given some knowledge on their distributions.

Depending on the application scenario, it can be desirable to reject both OOD samples and samples for which a correct class exist, but for which the classifier makes an error, be it due to an adversarial perturbation, class-preserving distributional shifts or simply standard test errors (Stutz et al., 2020; Sheikholeslami et al., 2021; Prusa & Voracek, 2023; Tramer, 2022). While some approaches combine OOD and error detection (Narasimhan et al., 2023), many angles of their interaction and their relevance in various real-world situations, which potentially have very different requirements, desired behaviours and applicable trade offs, are still unexplored.

Abbreviations

AAUC	Adversarial AUC
ACET	Adversarial Confidence Enhancing Training
Acc.	Accuracy
AUC	AUROC
AUROC	Area Under the Receiver-Operator Characteristic Curve
BGC	Background Class
CEDA	Confidence Enhancing Data Augmentation
CLIP	Contrastive Language-Image Pre-Training
CNN	Convolutional Neural Network
DNN	Deep Neural Network
FGSM	Fast Gradient Sign Method
FPR	False Positive Rate
FPR@95%TPR	False Positive Rate at 95% True Positive Rate
GAUC	Guaranteed AUC
GOOD	Guaranteed Out-Of-Distribution Detection
IBP	Interval Bound Propagation
ID	In-Distribution
i.i.d.	Independent and Identically Distributed
MSP	Maximum Softmax Probability
NINCO	No ImageNet Class Objects
OE	Outlier Exposure
OSR	Open Set Recognition
OOD	Out-Of-Distribution
PGD	Projected Gradient Descent
ProoD	Provable Out-Of-Distribution Detection
ReLU	Rectified Linear Unit
RBF	Radial Basis Function
SOTA	State-Of-The-Art
TE	Test Error
TPR	True Positive Rate
ViT	Vision Transformer

Bibliography

- Aaronson, S. $P \neq NP$. *Open Problems in Mathematics*. Springer International Publishing, 2016. <https://www.scottaaronson.com/papers/pnp.pdf>.
- Andriushchenko, M. and Flammarion, N. Understanding and improving fast adversarial training. *NeurIPS*, 2020.
- Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: a query-efficient black-box adversarial attack via random search. *ECCV*, 2020a.
- Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: a query-efficient black-box adversarial attack via random search. *ECCV*, 2020b.
- Anil, C., Lucas, J., and Grosse, R. Sorting out Lipschitz function approximation. *ICML*, 2019.
- Arora, R., Basuy, A., Mianjyz, P., and Mukherjee, A. Understanding deep neural networks with rectified linear unit. *ICLR*, 2018.
- Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *ICML*, 2018a.
- Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *ICML*, July 2018b. <https://arxiv.org/abs/1802.00420>.
- Augustin, M., Meinke, A., and Hein, M. Adversarial robustness on in-and out-distribution improves explainability. *ECCV*, 2020.
- Azizmalayeri, M., Soltani Moakhar, A., Zarei, A., Zohrabi, R., Manzuri, M., and Rohban, M. H. Your out-of-distribution detection method is not robust! *NeurIPS*, 2022.
- Bartlett, P. and Wegkamp, M. H. Classification with a reject option using a hinge loss. *JMLR*, 9:1823–1840, 2008.
- Bendale, A. and Boulton, T. Towards open set deep networks. *CVPR*, 2016a.
- Bendale, A. and Boulton, T. E. Towards open set deep networks. *CVPR*, 2016b.
- Berrada, L., Dathathri, S., Dvijotham, K., Stanforth, R., Bunel, R., Uesato, J., Goyal, S., Kumar, M. P., et al. Make sure you’re unsure: A framework for verifying probabilistic specifications. *NeurIPS*, 2021a.

- Berrada, L., Dathathri, S., Dvijotham, K. D., Stanforth, R., Bunel, R. R., Uesato, J., Goyal, S., and Kumar, M. P. Make sure you're unsure: A framework for verifying probabilistic specifications. *NeurIPS*, 2021b. <https://openreview.net/forum?id=oAxm0Wz7Bv>.
- Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. *ECML/PKDD*, 2013.
- Birhane, A. and Prabhu, V. U. Large image datasets: A pyrrhic win for computer vision? *WACV*, pp. 1537–1547, 2021.
- Bishop, C. M. Novelty detection and neural network validation. *IEEE Proceedings-Vision, Image and Signal processing*, 141:217–222, 1994.
- Bitterwolf, J., Meinke, A., and Hein, M. Certifiably adversarially robust detection of out-of-distribution data. *NeurIPS*, 2020. <https://github.com/j-cb/GOOD>.
- Bitterwolf, J., Meinke, A., Augustin, M., and Hein, M. Breaking down out-of-distribution detection: Many methods based on OOD training data estimate a combination of the same core quantities. *ICML*, 2022. https://github.com/j-cb/Breaking_Down_OOD_Detection.
- Bitterwolf, J., Mueller, M., and Hein, M. In or out? Fixing ImageNet out-of-distribution detection evaluation. *ICML*, 2023. <https://github.com/j-cb/NINCO>.
- Bossard, L., Guillaumin, M., and Van Gool, L. Food-101 – mining discriminative components with random forests. *ECCV*, 2014.
- Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge University Press, 2004.
- Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *ICLR*, 2018. <https://openreview.net/forum?id=SyZIOGWCZ>.
- Carlini, N. and Wagner, D. A. Adversarial examples are not easily detected: Bypassing ten detection methods. *10th ACM Workshop on Artificial Intelligence and Security*, 2017. <https://api.semanticscholar.org/CorpusID:207599948>.
- Chandola, V., Banerjee, A., and Kumar, V. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3), 2009.
- Chen, J., Li, Y., Wu, X., Liang, Y., and Jha, S. Informative outlier matters: Robustifying out-of-distribution detection using outlier mining. *ECML/PKDD*, 2021.
- Cheng, C.-H., Nührenberg, G., and Ruess, H. Maximum resilience of artificial neural networks. *International Symposium on Automated Technology for Verification and Analysis*, 2017.

- Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. *CVPR*, 2014.
- Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. Emnist: Extending mnist to handwritten letters. *IJCNN*, 2017.
- Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified adversarial robustness via randomized smoothing. *ICML*, 2019.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. Natural language processing (almost) from scratch. *JMLR*, 2011.
- Croce, F. and Hein, M. A randomized gradient-free attack on relu networks. *GCPR*, 2018.
- Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *ICML*, 2020.
- Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. Robustbench: a standardized adversarial robustness benchmark. *NeurIPS*, 2021. <https://openreview.net/forum?id=SSKZPJct7B>.
- Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. *CVPR*, 2019.
- Dathathri, S., Dvijotham, K., Kurakin, A., Raghunathan, A., Uesato, J., Bunel, R. R., Shankar, S., Steinhart, J., Goodfellow, I., Liang, P. S., and Kohli, P. Enabling certification of verification-agnostic networks via memory-efficient semidefinite programming. *NeurIPS*, 2020.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. *CVPR*, 2009.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.
- Elsayed, G., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., and Sohl-Dickstein, J. Adversarial examples that fool both computer vision and time-limited humans. *NeurIPS*, 2018.
- Fang, Z., Li, Y., Lu, J., Dong, J., Han, B., and Liu, F. Is out-of-distribution detection learnable? *NeurIPS*, 2022. https://proceedings.neurips.cc/paper_files/paper/2022/file/f0e91b1314fa5eabf1d7ef6d1561ecec-Paper-Conference.pdf.
- Fellbaum, C. Wordnet: An electronic lexical database. *MIT Press*, 1998.

- Fellbaum, C. Wordnet. *The encyclopedia of applied linguistics*, 2012.
- Fort, S. Adversarial vulnerability of powerful near out-of-distribution detection. *arXiv preprint arXiv:2201.07012*, 2022.
- Fort, S., Ren, J., and Lakshminarayanan, B. Exploring the limits of out-of-distribution detection. *NeurIPS*, 2021. <https://openreview.net/forum?id=j5NrN8ffXC>.
- Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *ICML*, 2016.
- Galil, I., Dabbah, M., and El-Yaniv, R. A framework for benchmarking class-out-of-distribution detection and its application to ImageNet. *ICLR*, 2023. <https://openreview.net/forum?id=Iuubb9W6Jtk>.
- Gilmer, J., Ford, N., Carlini, N., and Cubuk, E. Adversarial examples are a natural consequence of test error in noise. *ICML*, 2019.
- Golan, I. and El-Yaniv, R. Deep anomaly detection using geometric transformations. *NeurIPS*, 2018.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *NeurIPS*, 2014.
- Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *ICLR*, 2015.
- Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T., and Kohli, P. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv:1810.12715*, 2018.
- Gowal, S., Rebuffi, S., Wiles, O., Stimberg, F., Calian, D. A., and Mann, T. A. Improving robustness using generated data. *NeurIPS*, 2021.
- Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. Your classifier is secretly an energy based model and you should treat it like one. *ICLR*, 2020. <https://openreview.net/forum?id=Hkxzx0NtDB>.
- Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. On calibration of modern neural networks. *ICML*, 2017.
- He, K., Zhang, X., , Ren, S., and Sun, J. Deep residual learning for image recognition. *CVPR*, pp. 770–778, 2016.
- Hein, M. and Andriushchenko, M. Formal guarantees on the robustness of a classifier against adversarial manipulation. *NeurIPS*, 2017.

- Hein, M., Andriushchenko, M., and Bitterwolf, J. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. *CVPR*, 2019. https://github.com/max-andr/relu_networks_overconfident.
- Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019.
- Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR*, 2017. <https://openreview.net/forum?id=Hkg4TI9xl>.
- Hendrycks, D., Mazeika, M., and Dietterich, T. Deep anomaly detection with outlier exposure. *ICLR*, 2019a. <https://github.com/hendrycks/outlier-exposure>.
- Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. Using self-supervised learning can improve model robustness and uncertainty. *NeurIPS*, 2019b.
- Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. *CVPR*, 2021.
- Hendrycks, D., Basart, S., Mazeika, M., Zou, A., Kwon, J., Mostajabi, M., Steinhardt, J., and Song, D. Scaling out-of-distribution detection for real-world settings. *ICML*, 2022.
- Huang, R. and Li, Y. Mos: Towards scaling out-of-distribution detection for large semantic space. *CVPR*, 2021.
- Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Adversarial examples are not bugs, they are features. *NeurIPS*, 2019.
- Ismail, A., Ahmad, S. A., Che Soh, A., Hassan, M. K., and Harith, H. H. MYNursingHome: A fully-labelled image dataset for indoor object classification. *Data in Brief*, 2020. doi: <https://doi.org/10.1016/j.dib.2020.106268>. <https://www.sciencedirect.com/science/article/pii/S2352340920311628>.
- Jordan, M., Lewis, U., and Dimakis, A. G. Provable certificates for adversarial examples: Fitting a ball in the union of polytopes. *NeurIPS*, 2019.
- Jovanović, N., Balunović, M., Baader, M., and Vechev, M. On the paradox of certified training. *TMLR*, 2022a.
- Jovanović, N., Balunovic, M., Baader, M., and Vechev, M. On the paradox of certified training. *TMLR*, 2022b. <https://openreview.net/forum?id=atJHLVyBi8>.
- Katz, G., Barrett, C., Dill, D., Julian, K., and Kochenderfer, M. Reluplex: An efficient smt solver for verifying deep neural networks. *CAV*, 2017.

- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *ICLR*, 2015.
- Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. *ICML*, 2021.
- Koner, R., Sinhamahapatra, P., Roscher, K., Günnemann, S., and Tresp, V. OODformer: Out-of-distribution detection transformer. *arXiv preprint arXiv:2107.08976*, 2021.
- Kopetzki, A.-K., Charpentier, B., Zügner, D., Giri, S., and Günnemann, S. Evaluating robustness of predictive uncertainty estimation: Are Dirichlet-based models reliable? *ICML*, 2021.
- Kotha, S., Brix, C., Kolter, Z., Zhang, H., et al. Provably bounding neural network preimages. *arXiv preprint arXiv:2302.01404*, 2023.
- Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kuznetsova, A., Rom, H., Uijlings, J., Popov, S., Kamali, S., Mallocci, M., Pont-Tuset, J., Veit, A., Belongie, S., Gomes, V., Gupta, A., Sun, C., Chechik, G., Cai, D., Feng, Z., Narayanan, D., and Murphy, K. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available at storage.googleapis.com/openimages/web/index.html*, 2017.
- Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. *ICCV vision workshop*, 2013.
- Kristiadi, A., Hein, M., and Hennig, P. Being Bayesian, Even Just a Bit, Fixes Overconfidence in ReLU Networks. *ICML*, 2020.
- Kristiadi, A., Hein, M., and Hennig, P. An infinite-feature extension for bayesian relu nets that fixes their asymptotic overconfidence. *NeurIPS*, 2021.
- Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. *NeurIPS*, 2012.
- Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial examples in the physical world. *Artificial intelligence safety and security*. 2018.
- Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *NeurIPS*, 2017.
- Lapin, M., Hein, M., and Schiele, B. Loss functions for top-k error: Analysis and insights. *CVPR*, 2016.

- Laptev, D., Savinov, N., Buhmann, J., and Pollefeys, M. TI-pooling: Transformation-invariant pooling for feature learning in convolutional neural networks. *CVPR*, 2016.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. A tutorial on energy-based learning, 2006. <http://yann.lecun.com/exdb/publis/orig/lecun-06.pdf>.
- Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified robustness to adversarial examples with differential privacy. *IEEE symposium on security and privacy (SP)*, 2019.
- Lee, K., Lee, H., Lee, K., and Shin, J. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *ICLR*, 2018a.
- Lee, K., Lee, H., Lee, K., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *NeurIPS*, 2018b. https://github.com/pokaxpoka/deep_Mahalanobis_detector.
- Leibig, C., Allken, V., Ayhan, M. S., Berens, P., and Wahl, S. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific Reports*, 7, 2017.
- Li, F.-F., Andreeto, M., Ranzato, M., and Perona, P. Caltech 101, 2022.
- Li, Y. and Vasconcelos, N. Background data resampling for outlier-aware classification. *CVPR*, 2020.
- Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. *ICLR*, 2018.
- Libbrecht, M. W. and Noble, W. S. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 2015.
- Liu, H. and Abbeel, P. Hybrid discriminative-generative training via contrastive learning. *arXiv preprint arXiv:2007.09070*, 2020.
- Liu, W., Wang, X., Owens, J., and Li, Y. Energy-based out-of-distribution detection. *NeurIPS*, 2020.
- Liu, Y., Chen, X., Liu, C., and Song, D. Delving into transferable adversarial examples and black-box attacks. *ICLR*, 2017.
- Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. *ICCV*, 2015.

- Lundervold, A. S. and Lundervold, A. An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift für Medizinische Physik*, 2019.
- Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Valdu, A. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018.
- Maier-Hein, L., Eisenmann, M., Sarikaya, D., März, K., Collins, T., Malpani, A., Fallert, J., Feussner, H., Giannarou, S., Mascagni, P., et al. Surgical data science—from concepts toward clinical translation. *Medical image analysis*, 76:102306, 2022.
- Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. Fine-grained visual classification of aircraft. *arXiv:1306.5151*, 2013.
- Malinin, A. and Gales, M. Predictive uncertainty estimation via prior networks. *NeurIPS*, 2018.
- Malinin, A. and Gales, M. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. *NeurIPS*, 2019.
- Meinke, A. and Hein, M. Towards neural networks that provably know when they don’t know. *ICLR*, 2020. <https://github.com/AlexMeinke/certified-certain-uncertainty>.
- Meinke, A., Bitterwolf, J., and Hein, M. Provably robust detection of out-of-distribution data (almost) for free. *NeurIPS*, 2022. <https://github.com/AlexMeinke/Provable-OOD-Detection>.
- Ming, Y., Cai, Z., Gu, J., Sun, Y., Li, W., and Li, Y. Delving into out-of-distribution detection with vision-language representations. *NeurIPS*, 2022. <https://openreview.net/forum?id=KnCS9390Va>.
- Mirman, M., Gehr, T., and Vechev, M. Differentiable abstract interpretation for provably robust neural networks. *ICML*, 2018.
- Mohseni, S., Pitale, M., Yadawa, J., and Wang, Z. Self-supervised learning for generalizable out-of-distribution detection. *AAAI*, 2020.
- Montufar, G., Pascanu, R., Cho, K., and Bengio, Y. On the number of linear regions of deep neural networks. *NeurIPS*, 2014.
- Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Hybrid models with deep and invertible features. *ICML*, 2019.
- Nalisnick, E., Matsukawa, A., Whye Teh, Y., Gorur, D., and Lakshminarayanan, B. Do deep generative models know what they don’t know? *ICLR*, 2019.
- Narasimhan, H., Menon, A. K., Jitkrittum, W., and Kumar, S. Learning to reject meets OOD detection: Are all abstentions created equal? *arXiv preprint arXiv:2301.12386*, 2023.

- Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.
- Nguyen, A., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *CVPR*, 2015.
- Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 2008.
- Papadopoulos, A.-A., Rajati, M. R., Shaikh, N., and Wang, J. Outlier exposure with confidence control for out-of-distribution detection. *Neurocomputing*, 2021.
- Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. *AsiaCCS*, 2017.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011.
- Peng, S., Xu, W., Cornelius, C., Hull, M., Li, K., Duggal, R., Phute, M., Martin, J., and Chau, D. H. Robust principles: Architectural design principles for adversarially robust cnns. *arXiv preprint arXiv:2308.16258*, 2023.
- Prusa, D. and Voracek, V. Optimal strategies for reject option classifiers. *JMLR*, 2023.
- Qin, C., Martens, J., Gowal, S., Krishnan, D., Dvijotham, K., Fawzi, A., De, S., Stanforth, R., and Kohli, P. Adversarial robustness through local linearization. *NeurIPS*, 32, 2019.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. *ICML*, 2021.
- Raghunathan, A., Xie, S. M., Yang, F., Duchi, J., and Liang, P. Adversarial training can hurt generalization. *ICML 2019 Workshop on Identifying and Understanding Deep Learning Phenomena*, 2019.
- Rashid, A., Hacker, S., Zhang, G., Kristiadi, A., and Poupart, P. Preventing arbitrarily high confidence on far-away data in point-estimated discriminative neural networks. *AISTATS*, 2024.
- Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., DePristo, M. A., Dillon, J. V., and Lakshminarayanan, B. Likelihood ratios for out-of-distribution detection. *NeurIPS*, 2019.

- Ren, J., Fort, S., Liu, J., Roy, A. G., Padhy, S., and Lakshminarayanan, B. A simple fix to mahalanobis distance for improving near-ood detection, 2021. <https://arxiv.org/abs/2106.09022>.
- Rice, L., Wong, E., and Kolter, Z. Overfitting in adversarially robust deep learning. *ICML*, 2020.
- Roy, A. G., Ren, J. J., Azizi, S., Loh, A., Natarajan, V., Mustafa, B., Pawlowski, N., Freyberg, J., Liu, Y., Beaver, Z. W., Vo, N., Bui, P., Winter, S., MacWilliams, P., Corrado, G., Telang, U., Liu, Y., Cemgil, T., Karthikesalingam, A., Lakshminarayanan, B., and Winkens, J. Does your dermatology classifier know what it doesn't know? detecting the long-tail of unseen conditions. *Medical Imaging Analysis*, 2021.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.
- Salehi, M., Mirzaei, H., Hendrycks, D., Li, Y., Rohban, M. H., and Sabokrou, M. A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *TMLR*, 2021.
- Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., and Boulton, T. E. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2012.
- Sehwag, V., Bhagoji, A. N., Song, L., Sitawarin, C., Cullina, D., Chiang, M., and Mittal, P. Better the devil you know: An analysis of evasion attacks using out-of-distribution adversarial examples. *preprint, arXiv:1905.01726*, 2019.
- Sensoy, M., Kaplan, L., and Kandemir, M. Evidential deep learning to quantify classification uncertainty. *NeurIPS*, 2018.
- Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. Adversarial training for free! *NeurIPS*, 2019.
- Shankar, V., Roelofs, R., Mania, H., Fang, A., Recht, B., and Schmidt, L. Evaluating machine accuracy on ImageNet. *ICML*, 2020.
- Sheikholeslami, F., Lotfi, A., and Kolter, J. Z. Provably robust classification of adversarial examples with detection. *ICLR*, 2021. <https://openreview.net/forum?id=sRA5rLNpmQc>.
- Song, Y., Sebe, N., and Wang, W. Rankfeat: Rank-1 feature removal for out-of-distribution detection. *NeurIPS*, 2022.

- Steiner, A. P., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., and Beyer, L. How to train your ViT? data, augmentation, and regularization in vision transformers. *TMLR*, 2022. <https://openreview.net/forum?id=4nPswr1KcP>.
- Stutz, D., Hein, M., and Schiele, B. Disentangling adversarial robustness and generalization. *CVPR*, 2019.
- Stutz, D., Hein, M., and Schiele, B. Confidence-calibrated adversarial training: Generalizing robustness to unseen attacks. *ICML*, 2020.
- Sun, Y., Guo, C., and Li, Y. React: Out-of-distribution detection with rectified activations. *NeurIPS*, 2021.
- Sun, Y., Ming, Y., Zhu, X., and Li, Y. Out-of-distribution detection with deep nearest neighbors. *ICML*, 2022.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *ICLR*, 2014.
- Tack, J., Mo, S., Jeong, J., and Shin, J. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *NeurIPS*, 2020.
- Techapanurak, E., Suganuma, M., and Okatani, T. Hyperparameter-free out-of-distribution detection using cosine similarity. *Proceedings of the Asian Conference on Computer Vision*, 2020.
- Thulasidasan, S., Thapa, S., Dhaubhadel, S., Chennupati, G., Bhattacharya, T., and Bilmes, J. An effective baseline for robustness to distributional shift. *International Conference on Machine Learning and Applications*, 2021.
- Tjeng, V. and Tedrake, R. Verifying neural networks with mixed integer programming. preprint, arXiv:1711.07356v1, 2017.
- Torralba, A., Fergus, R., and Freeman, W. T. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE PAMI*, 2008.
- Touvron, H., Cord, M., and Jegou, H. DeiT III: Revenge of the ViT. *arXiv preprint arXiv:2204.07118*, 2022.
- Tramer, F. Detecting adversarial examples is (nearly) as hard as classifying them. *ICML*, 2022.
- Tramer, F., Carlini, N., Brendel, W., and Madry, A. On adaptive attacks to adversarial example defenses. *NeurIPS*, 2020.
- Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. *ICLR*, 2019. <https://openreview.net/forum?id=SyxAb30cY7>.

- Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. The iNaturalist species classification and detection dataset. *CVPR*, 2018.
- Vasudevan, V., Caine, B., Gontijo-Lopes, R., Fridovich-Keil, S., and Roelofs, R. When does dough become a bagel? Analyzing the remaining mistakes on ImageNet. Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *NeurIPS*, 2022. <https://openreview.net/forum?id=mowt1WNhTC7>.
- Vaze, S., Han, K., Vedaldi, A., and Zisserman, A. Open-set recognition: a good closed-set classifier is all you need? *ICLR*, 2022.
- Veerabadran, V., Goldman, J., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., Shlens, J., Sohl-Dickstein, J., Mozer, M. C., et al. Subtle adversarial image manipulations influence both human and machine perception. *Nature Communications*, 2023.
- Wang, H., Liu, W., Bocchieri, A., and Li, Y. Can multi-label classification networks know what they don’t know? *NeurIPS*, 2021.
- Wang, H., Li, Z., Feng, L., and Zhang, W. ViM: Out-of-distribution with virtual-logit matching. *CVPR*, 2022a.
- Wang, H., Zhang, A., Zhu, Y., Zheng, S., Li, M., Smola, A., and Wang, Z. Partial and asymmetric contrastive learning for out-of-distribution detection in long-tailed recognition. *ICML*, 2022b.
- Wen, X., Zhao, B., and Qi, X. Parametric classification for generalized category discovery: A baseline study. *ICCV*, 2023.
- Wicker, M., Laurenti, L., Patane, A., and Kwiatkowska, M. Probabilistic safety for bayesian neural networks. *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020.
- Wightman, R. Pytorch image models, 2019. <https://github.com/rwightman/pytorch-image-models>.
- Winkens, J., Bunel, R., Roy, A. G., Stanforth, R., Natarajan, V., Ledsam, J. R., MacWilliams, P., Kohli, P., Karthikesalingam, A., Kohl, S., et al. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*, 2020.
- Wong, E. and Kolter, J. Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. *ICML*, 2018.
- Wong, E., Rice, L., and Kolter, J. Z. Fast is better than free: Revisiting adversarial training. *ICML*, 2019.

- Xia, G. and Bouganis, C.-S. On the usefulness of deep ensemble diversity for out-of-distribution detection. *arXiv preprint arXiv:2207.07517*, 2022.
- Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. preprint, arXiv:1708.07747, 2017.
- Xiao, Z., Yan, Q., and Amit, Y. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. *NeurIPS*, 2020.
- Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. Self-training with noisy student improves ImageNet classification. *CVPR*, 2020.
- Xiong, H. Y., Alipanahi, B., Lee, L. J., Bretschneider, H., Merico, D., Yuen, R. K., Hua, Y., Gueroussov, S., Najafabadi, H. S., Hughes, T. R., et al. The human splicing code reveals new insights into the genetic determinants of disease. *Science*, 2015.
- Yang, J., Wang, P., Zou, D., Zhou, Z., Ding, K., Peng, W., Wang, H., Chen, G., Li, B., Sun, Y., et al. Openood: Benchmarking generalized out-of-distribution detection. *NeurIPS*, 2022.
- Yang, W., Zhang, B., and Russakovsky, O. ImageNet-OOD: Deciphering modern out-of-distribution detection algorithms. *arXiv preprint arXiv:2310.01755*, 2023. <https://doi.org/10.48550/arXiv.2310.01755>.
- Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.
- Zagoruyko, S. and Komodakis, N. Wide residual networks. *BMVC*, 2016.
- Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. *ICML*, 2019.
- Zhang, H., Chen, H., Xiao, C., Goyal, S., Stanforth, R., Li, B., Boning, D., and Hsieh, C.-J. Towards stable and efficient training of verifiably robust neural networks. *ICLR*, 2020.
- Zhang, J., Yang, J., Wang, P., Wang, H., Lin, Y., Zhang, H., Sun, Y., Du, X., Zhou, K., Zhang, W., Li, Y., Liu, Z., Chen, Y., and Li, H. OpenOOD v1.5: Enhanced benchmark for out-of-distribution detection. *arXiv preprint arXiv:2306.09301*, 2023.
- Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.
- Zhou, S. K., Greenspan, H., Davatzikos, C., Duncan, J. S., Van Ginneken, B., Madabhushi, A., Prince, J. L., Rueckert, D., and Summers, R. M. A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proceedings of the IEEE*, 109(5):820–838, 2021.