Enhancement and Evaluation of Deep Generative Networks with Applications in Super-Resolution and Image Generation

Enhancement and Evaluation of Deep Generative Networks with Applications in Super-Resolution and Image Generation

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. Seyed Mohammad Mehdi Sajjadi
aus Hamburg

Tübingen
2024

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

Tag der mündlichen Qualifikation:	04.07.2024
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Hendrik P. A. Lensch
2. Berichterstatter:	Prof. Dr. Bernhard Schölkopf

Abstract

Since the advent of computers, perceiving the world visually has been a major focus of research. Today, raster graphics are the most popular format for storing arbitrary visual data. This choice of representation carries major advantages, primarily high flexibility and suitability for hardware acceleration. However, increasing the size of images leads to well-known blurring or pixelization artifacts. The field of super resolution (SR) investigates methods to improve the quality of enlarged visual data.

This work is structured into two sections: making strides towards more realistic and efficient SR, and improving the architecture and evaluation of deep generative models that form the foundation on which today's best SR methods are built. While prior art has primarily focused on improving network architechtures of deep neural networks, we propose a shift in focus to the loss functions used to train the models. We present *EnhanceNet*, a novel method that achieves state-of-the-art quantitative and qualitative image quality in SR through a novel set of training objectives. A combination of perceptual, style, and adversarial networks applied to the task of SR leads to previously unattainable visual fidelity at large scaling factors. Extending image SR methods to video data is regularly achieved through feeding a number of neighbouring frames into a neural network that is applied in a sliding window across time. The major shortcoming of this common approach lies in its low computational efficiency as each frame is processed independently several times, and the resulting temporal instabilities in the outputs. We propose *Frame-Recurrent Video Super Resolution*, a method that recurrently uses the output of the last frame to upsample the next one. The method achieves state-of-the-art video SR quality while vastly improving computational requirements and temporal consistency.

GANs are a method as powerful, as difficult to train, regularly suffering from failures due to bad gradients. We propose *Tempered Adversarial Networks*, a novel way to auto-stabilize GAN training through the introduction of a lens module that modifies real data samples to look more similar to generated ones throughout training. A range of experiments shows the promise of such techniques in improving gradients for the generator and therefore improving success rates of training. Measuring the success of these methods is known to be a challenging task, as it implies matching distributions rather than pairs of samples. We define *Precision and Recall for Distributions* which disentangles a measure of quality of samples from coverage of the original data distribution. We close with *Regularized AutoEncoders*, a study into differences and similarities between Auto Encoders, VAEs and several forms inbetween. The major finding is that stochastic VAEs are not always required for the task they are set out to solve, and simpler RAEs often outperform their stochastic counterparts.

Kurzfassung

Seit dem Beginn von Computern ist die visuelle Wahrnehmung der Welt einer der wichtigsten Forschungsschwerpunkte. Während Rastergrafiken das üblichste Format zur Verarbeitung beliebiger visueller Daten sind, führt die Vergrößerung von Bildern zur bekannten Verpixelung dieser. Das Gebiet der Super Resolution (SR) untersucht Methoden zur Verbesserung der Qualität von vergrößerten Bildern und Videos.

Diese Arbeit ist in zwei Abschnitte gegliedert: Fortschritte zu realistischer und effizienter SR, sowie die Verbesserung der Architektur und Evaluation generativer Modelle, auf denen die aktuell besten SR-Methoden basieren.

Während sich die meiste Forschung hauptsächlich auf die Verbesserung von Netzwerkarchitekturen tiefer neuronaler Netzwerke konzentriert haben, schlagen wir einen Schwerpunktwechsel hin zu Losses vor, die zum Optimieren der Modelle verwendet werden. Wir präsentieren *EnhanceNet*, eine Methode, die durch eine neuartige Kombination aus perzeptueller, stilistischer und sog. *adversarial* Losses eine bisher unerreichte Bildqualität in der SR erreicht.

Die Anwendung von SR-Methoden auf Videodaten wird oft durch das Einspeisen einer bestimmten Anzahl benachbarter Videobildern in ein neuronales Netzwerk erreicht, was nicht nur ineffizient ist, sondern oftmals zum Flickern in der Ausgabe führt. Wir präsentieren *Frame-Recurrent Video Super Resolution*, eine Methode, die die Ausgabe für das vorherige Videobild rekurrent weiterverwendet, um das nächste zu vergrößern. Die Methode übertrifft damit sämtliche bisherige Video SR Methoden nicht nur in Bildqualität, sondern verbessert auch die Rechenanforderungen und temporale Konsistenz.

GANs sind eine ebenso mächtige, wie schwer zu optimierende Methode, die häufig aufgrund schlechter Gradienten zum Totalversagen führt. Wir präsentieren *Tempered Adversarial Networks*, eine neuartige Methode zur automatischen Stabilisierung des Trainings durch die Einführung eines dritten Moduls, welches reale Datenpunkte so modifiziert, dass sie den generierten ähnlicher aussehen. Eine Reihe von Experimenten belegt das Potenzial dieser Technik zur Stabilisierung des Trainings.

Die Evaluation solcher Methoden gilt als eine extrem herausfordernde Aufgabe. Wir präsentieren *Precision and Recall for Distributions*, eine neue Evaluationsmetrik, die die Qualität von Daten von ihrer Abdeckung der Originalverteilung entkoppelt, und damit eine genauere Evaluation von generativen Modellen erlaubt.

Zum Abschluss präsentieren wir *Regularized Autoencoders* (RAE), eine Untersuchung von Autoencodern, VAEs, und mehrerer Zwischenformen. Die Haupterkenntnis ist, dass VAEs in der Praxis nicht immer erforderlich sind, sondern dass simple RAE oft ihre stochastischen Gegenstücke übertreffen.

Acknowledgments

First and foremost, I would like to thank Bernhard Schölkopf for offering me the opportunity to work with him throughout these years. I wholeheartedly enjoyed the time, and the studies in Tübingen have shaped me personally and professionally in distinctive ways that would only have been possible in this unique setting. I want to express my heartfelt gratitude to Ulrike von Luxburg for imparting to me early the means to principled research. I also thank her group members Morteza Alamgir, Siavash Haghiri, and Matthäus Kleindessner for advising me early on. None of my work would have been possible without each and every one of my amazing coauthors, who I thank for the countless whiteboard discussions, coding sessions, late-night deadline pushes, philosophical debates, and further fruitful exchanges. My thanks extend to the entire group of Empirical Inference, in particular Chaochao Lu, Alexander Neitz, Patrick Wieschollek and Matthias Bauer. I further thank the many members of the other departments and groups at the MPI for making my stay a warm one. I am particularly grateful to Michael Black's Perceiving Systems group who welcomed me as a cherished unofficial member and made my every day and night an absolute delight. Thank you Partha, Anurag, Soubhik, Priyanka, Timo, Haiwen, Ahmed, Nima, and everyone else – you're all awesome. I am grateful to my hosts and colleagues Raviteja Vemulapalli, Matthew Brown, Mario Lučić, Sylvain Gelly, Olivier Bachem, and Olivier Bousquet for my invaluable internships at Google. A special thanks goes out to Patrick Putzky for rekindling my passion for board games, Zachariah Henseler and Daphne Welter for their constant positivity, and all further members of the regular gatherings. I sincerely thank my family who I owe everything to, for supporting me in innumerable and indescribable ways. I cannot thank enough my closest friends that I have shared some of the best time of my life with. Thank you Finn, Lars, Partha, Chantal, Tobias. I am grateful for the ETH CLS programme that accepted me as an Associated PhD Fellow. I sincerely thank my dissertation committee Hendrik Lensch and Bernhard Schölkopf for reviewing my work. I especially thank the researchers I talked to at my first major conference, ICML 2015 in Lille, which led me to pursue a PhD in machine learning. And finally, I thank everyone who I interacted with during my studies, shaping me into the person I am today.

Contents

Chapter 1

Introduction

Image reconstruction has a long history in computer vision due to its numerous applications. In its purest form, the task of image reconstruction is to simply improve image quality. There are a number of different subfields such as denoising, deblurring, demosaicing, defencing and dehazing. This thesis contains advances in deep neural networks for image and video reconstruction on the one hand, and deep generative models for data modeling and synthesis on the other hand. For image reconstruction, we focus on the task of super-resolution, though all proposed methods can be easily extended to further tasks.

A flexible and one of the most commonly used representations of images in computer systems are raster graphics. Raster graphics allow full flexibility over the contents of the image, however they have a severe limitation when it comes to enlarging images: they will look pixelated when zoomed in, and commonly used methods such as Bicubic interpolation lead to blurry results that lack details and are unpleasing to the eye. This need has sparked a great interest in the research of super-resolution which studies methods of increasing the resolution of images and videos. While classical methods based on signal processing lead to mediocre results, recent advances in machine learning and especially in deep learning have significantly improved popular benchmarks over the years.

Chapter 2 cites Sajjadi *et al.* (2017) which has been published and orally presented at ICCV 2017. We investigate the question of the correct loss function to use for the task of single image super-resolution. Almost all prior work in the field of single image super-resolution has focused on finding techniques to more efficiently handle the data and on the design of deeper network architectures while neglecting the importance of the loss function needed to achieve a higher perceived image quality. While the common choice for a training objective is the mean squared error (MSE), it is known to lead to blurry images and washed out textures. Similarly, the most commonly used evaluation metrics PSNR and SSIM in fact encourage blurrier results since they are more accomodating of missing alignment between the generated and ground truth imagery. We investigate the shortcomings of the MSE loss for the task of super-resolution and propose to use a combination of the perceptual loss, texture synthesis loss, and an adversarial loss to achieve sharper, more detailed images with realistic textures. The findings are evaluated in an extensive qualitative and quantitative study, and we show that the results of the proposed method greatly outperforms prior works even when our method is put at a strong

disadvantage to the competition.

A video is defined as an ordered tuple of frames which are generally temporally coherent, *i.e.*, neighboring frames are assumed to be similar in content. While the task of video super-resolution may seem like a similar problem to single image super-resolution, it opens up some new opportunities and pitfalls. The key to accurate video super-resolution lies in exploiting its temporal smoothness across nearby video frames, making use of the fact that further views of the same content are often available. Consequently, video super-resolution methods have to combine the information between subsequent frames to achieve the best results. Meanwhile, video super-resolution introduces the new challenge of temporal consistency. Natural videos exhibit smooth temporal statistics, *i.e.*, the value of a pixel does not change by a large value from one frame to the next. Super-resolving videos in a way that leads to temporally consistent results is crucial for methods in this field since violations thereof lead to flickering artifacts, rendering the video unpleasing to the human eye.

Chapter 3 focuses on the task of efficient, high-quality, and temporally consistent video super-resolution, citing from Sajjadi *et al.* (2018b) published at CVPR 2018. To combine information from neighboring frames for the generation of the current frame, previous works have posed the problem as a multi-frame super-resolution task. To super-resolve the current frame I_t, a select number of neighboring frames is aligned with the current frame and all images are finally stacked and passed through a super-resolution network. This procedure is applied to all frames over the video independently in a sliding window fashion. We identify two main problems with this approach that lead to shortcomings in terms of quality, runtime efficiency and temporal consistency. The first shortcoming is that each input frame is processed several times as the sliding window passes over the video with no information sharing, leading to subpar efficiency as each frame should ideally be processed only once. The second issue is that each *output* frame is generated independently, making it challenging for the super-resolution network to produce temporally consistent results as it cannot reuse previous computation.

We propose to solve both shortcomings with a frame-recurrent approach. Instead of passing a sliding window over the video, we process each input frame only once, and we instead align the previously generated high-resolution output frame with the current input. This has the advantage that the network only needs to add the new information from the current input frame to the previously generated high-resolution image, *i.e.* there is a lower computational burden on the network. At the same time, the network directly sees the previously generated frame, so it has no constraints when it comes to generating temporally consistent videos. A comprehensive evaluation study and comparison of the proposed frame-recurrent framework with previous works shows a substantial boost in video quality, temporal consistency, and runtime efficiency.

The remaining chapters cover advances in deep generative modeling with a particular focus on high-dimensional complex datasets such as images. The overall goal of generative modeling is to capture the full probability distribution that has generated the dataset given through a commonly finite set of samples. In recent years, deep neural networks have

made great strides towards higher-quality unsupervised image generation. Two of the most popular models for this task are *generative adversarial networks* (GANs) and *variational autoencoders* (VAEs). GANs are very powerful models that have led to state-of-the-art results on unsupervised image generation on the scale of varied natural images. A GAN consists of a generator that produces fake samples and a discriminator which is trained to distinguish fake, *i.e*. generated, samples from real samples from the original dataset. Training GANs thus involves a min-max optimization problem which has been found to be very unstable, *i.e*., the training process frequently breaks down and a large amount of hyperparameter tuning is necessary for good results. A further shortcoming of GANs is mode drop: the original data distribution often consists of several modes (*e.g*., images of different classes of objects), and while GANs regularly create realistically looking samples, they tend to drop some of the modes available in the original dataset.

Chapter 4 cites from Sajjadi *et al.* (2018c), where we investigate the training instability problems of GANs. The intuition behind the proposed approach stems from the idea that it is intuitively harder to generate realistic samples than to tell them apart from real samples. This imbalance leads to the discriminator *overpowering* the generator during training, *i.e*., it achieves perfect accuracy in telling real samples from fakes ones. This in turn leads to uninformative and exploding training gradients for the generator, resulting in the ubiquitous training failures of GANs. We propose to tackle the problems above by inverting the training process. A so-called *lens* module is added between the real data and the discriminator which is trained to make the real data look like the generated data throughout training. As the generator produces samples of increasing quality, the lens reduces its pertubations on the real data, such that together, the lens and the generator produce samples of better quality while fully avoiding the situation where the discriminator has a too easy job of telling real samples apart from fake ones. In an extensive evaluation section, we show that this simple addition to the GAN framework makes training more stable and ultimately leads to higher-quality samples of more variety.

To be able to fairly and accurately compare generative models, there is a need for quantitative methods of model evaluation. For this purpose, several approaches have been proposed with a varying degree of success. The Fréchet Inception Distance (FID) is one of the most popular choices since it has been found to correlate well with the perceived quality and variety of the generated samples. In Chapter 5, citing from Sajjadi *et al.* (2018a) which has been published at NeurIPS 2018, we show that not just the FID, but in fact *any* one-dimensional evaluation metric (*i.e*., the result of the evaluation is a scalar score or distance) has the severe limitation that it cannot separate sample quality from sample variety, or coverage of the true data distribution. To this end, we propose a novel evaluation method which disentangles the quality of samples (*precision*) from *recall*, which measures how much of the target distribution is captured by the generative model. We formally propose a definition of precision and recall for distributions (PRD) and prove that the definition is sound and has good properties. Since the definition is very general, finding a way to compute the set of solutions is nontrivial. However, we show that a surprisingly simple and efficient algorithm can compute PRD in the theoretical setting

where the densities are known. For the application in the real world (*i.e.*, to evaluate deep generative models), we propose a pipeline that leads to estimates of the PRD for a given set of samples without a need for having the probability density of the real or generated distributions. Finally, we evaluate the PRD of 800 generative models including GANs and VAEs on four datasets. The results convincingly show that our method successfully dintinguishes models that produce higher-quality samples from models that generate a large variety of samples.

Finally, Chapter 6 addresses the Variational Autoencoder (VAE) and some of its shortcomings, citing from Ghosh* *et al.* (2019). VAEs are a popular alternative to GANS for deep generative models due to their more stable training behavior and theoretically backed framework. However, VAEs tend to produce blurry images, an effect that is linked to their training objective and in part to the aggregated posterior mismatch. We invesigate the effects of the stochasticity in the VAE and formally show that the sampling process in the latent space is equivalent to simple regularization techniques based on the injection of Gaussian noise. With this in mind, we propose an alternative deterministic model, the Regularized Autoencoder (RAE), which replaces the implicit regularization with explicit smoothing techniques such as the gradient penalty and spectral normalization. In conjunction to this change, we further propose an efficient ex-post density estimation that yields a generative model that outperforms the VAE and even more sophisticated alternatives such as the Wasserstein Autoencoder. We finally show on several datasets that the simpler RAE framework achieves state-of-the-art results among autoencoding frameworks on standard datasets such as CelebA.

Chapter 2

EnhanceNet: Super-Resolution Through Automated Texture Synthesis

Single image super-resolution is the task of inferring a high-resolution image from a single low-resolution input. Traditionally, the performance of algorithms for this task is measured using pixel-wise reconstruction measures such as peak signal-to-noise ratio (PSNR) which have been shown to correlate poorly with the human perception of image quality. As a result, algorithms minimizing these metrics tend to produce over-smoothed images that lack high-frequency textures and do not look natural despite yielding high PSNR values.

We propose a novel application of automated texture synthesis in combination with a perceptual loss focusing on creating realistic textures rather than optimizing for a pixel-accurate reproduction of ground truth images during training. By using feed-forward fully convolutional neural networks in an adversarial training setting, we achieve a significant boost in image quality at high magnification ratios. Extensive experiments on a number of datasets show the effectiveness of our approach, yielding state-of-the-art results in both quantitative and qualitative benchmarks.

2.1 Introduction

Enhancing and recovering a high-resolution (HR) image from a low-resolution (LR) counterpart is a theme both of science fiction movies and of the scientific literature. In the latter, it is known as single image super-resolution (SISR), a topic that has enjoyed much attention and progress in recent years. The problem is inherently ill-posed as no unique solution exists: when downsampled, a large number of different HR images can give rise to the same LR image. For high magnification ratios, this one-to-many mapping problem becomes worse, rendering SISR a highly intricate problem. Despite considerable progress in both reconstruction accuracy and speed of SISR, current state-of-the-art methods are still far from image enhancers like the one operated by Harrison Ford alias Rick Deckard in the iconic Blade Runner movie from 1982. A crucial problem is the loss of high-frequency information for large downsampling factors rendering textured

ENet-E: State of the art by PSNR ENet-PAT: Our best result

Figure 2.1: Comparing the new state of the art by PSNR, ENet-E, with the sharper and perceptually more plausible result produced by ENet-PAT at 4x super-resolution. While the PSNR of the image on the right is lower, it is richer in texture, missing the typical dreamy look of super-resolved images such as the one on the left.

regions in super-resolved images blurry, overly smooth, and unnatural in appearance (*c.f.* Figure 2.1, left).

The reason for this behavior is rooted in the choice of the objective function that current state-of-the-art methods employ: most systems minimize the pixel-wise mean squared error (MSE) between the HR ground truth image and its reconstruction from the LR observation, which has however been shown to correlate poorly with human perception of image quality (Wang *et al.*, 2004; Laparra *et al.*, 2016). While easy to minimize, the optimal MSE estimator returns the mean of many possible solutions which makes SISR results look unnatural and implausible (*c.f.* Figure 2.2). This regression-to-the-mean problem in the context of super-resolution is a well-known fact, however, modeling the high-dimensional multi-modal distribution of natural images remains a challenging problem.

In this work we pursue a different strategy to improve the perceptual quality of SISR results. Using a fully convolutional neural network architecture, we propose a novel modification of recent texture synthesis networks in combination with adversarial training and perceptual losses to produce realistic textures at large magnification ratios. The method works on all RGB channels simultaneously and produces sharp results for natural images at a competitive speed. Trained with suitable combinations of losses, we reach state-of-the-art results both in terms of PSNR and using perceptual metrics.

2.2 Related Work

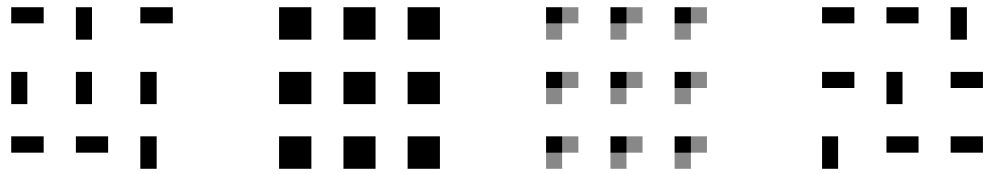
The task of SISR has been studied for decades (Irani and Peleg, 1991). Early interpolation methods such as bicubic and Lanczos (Duchon, 1979) are based on sampling theory but often produce blurry results with aliasing artifacts in natural images. A large number of high-performing algorithms have since been proposed (Milanfar, 2010), see also the recent surveys Nasrollahi and Moeslund (2014) and Yang *et al.* (2014).

In recent years, popular approaches include exemplar-based models that either exploit recurrent patches of different scales within a single image (Glasner *et al.*, 2009; Yang *et al.*, 2010a; Freedman and Fattal, 2011; Huang *et al.*, 2015a) or learn mappings between low and high resolution pairs of image patches in external databases (Freeman *et al.*, 2002; Chang *et al.*, 2004; Kim and Kwon, 2010; Bevilacqua *et al.*, 2012; Yang *et al.*, 2013; Yue *et al.*, 2013; Timofte *et al.*, 2014). They further include dictionary-based methods (Yang *et al.*, 2010b; Lu *et al.*, 2012; Zhang *et al.*, 2012; Yang *et al.*, 2012; Perez-Pellitero *et al.*, 2016; Timofte *et al.*, 2016) that learn a sparse representation of image patches as a combination of dictionary atoms, as well as neural network-based approaches (Dong *et al.*, 2014; Kim *et al.*, 2016b,a; Shi *et al.*, 2016a; Bruna *et al.*, 2016; Johnson *et al.*, 2016; Shi *et al.*, 2016b; Yu and Porikli, 2016; Dong *et al.*, 2016) which apply convolutional neural networks (CNNs) to the task of SISR. Some approaches are specifically designed for fast inference times (Perez-Pellitero *et al.*, 2016; Romano *et al.*, 2016; Shi *et al.*, 2016a). Thus far, realistic textures in the context of high-magnification SISR have only been achieved by user-guided methods (Tai *et al.*, 2010; HaCohen *et al.*, 2010).

More specifically, Dong *et al.* (2014) apply shallow networks to the task of SISR by training a CNN via backpropagration to learn a mapping from the bicubic interpolation of the LR input to a high-resolution image. Later works successfully apply deeper networks and the current state of the art in SISR measured by PSNR is based on deep CNNs (Kim *et al.*, 2016b,a).

As these models are trained through MSE minimization, the results tend to be blurry and lack high-frequency textures due to the afore-mentioned regression-to-the-mean problem. Alternative perceptual losses have been proposed for CNNs (Dosovitskiy and Brox, 2016; Johnson *et al.*, 2016) where the idea is to shift the loss from the image-space to a higher-level feature space of an object recognition system like VGG (Simonyan and Zisserman, 2015), resulting in sharper results despite lower PSNR values.

CNNs have also been found useful for the task of texture synthesis (Gatys *et al.*, 2015) and style transfer (Johnson *et al.*, 2016; Gatys *et al.*, 2016; Ulyanov *et al.*, 2016), however these methods are constrained to the setting of a single network learning to produce only a single texture and have so far not been applied to SISR. Adversarial networks (Goodfellow *et al.*, 2014) have recently been shown to produce sharp results in a number of image generation tasks (Denton *et al.*, 2015; Radford *et al.*, 2016a; Pathak *et al.*, 2016; Zhu *et al.*, 2016) but have so far only been applied in the context of super-resolution in a highly constrained setting for the task of face hallucination (Yu and Porikli, 2016).



(I) HR image I_{HR} | (II) LR image I_{LR} | (III) I_{est} optimal MSE | (IV) I_{est} adversarial loss

Figure 2.2: Toy example to illustrate the effect of the Euclidean loss and how maximizing the PSNR does not lead to realistic results. (I) The HR images consist of randomly placed vertical and horizontal bars of 1×2 pixels. (II) In I_{LR}, the original orientations cannot be distinguished anymore since both types of bars turn into a single pixel. (III) A model trained to minimize the Euclidean loss produces the mean of all possible solutions since this yields the lowest MSE but the result looks clearly different from the original images I_{HR}. (IV) Training a model with an adversarial loss ideally results in a sharp image that is impossible to distinguish from the original HR images, although it does not match I_{HR} exactly since the model cannot know the orientation of each bar. Intriguingly, this result has a lower PSNR than the blurry MSE sample.

2.3 Single Image Super-Resolution

A high resolution image $I_{\text{HR}} \in [0,1]^{\alpha H \times \alpha W \times 3}$ is downsampled to a low resolution image

$$I_{\text{LR}} = \text{d}_\alpha(I_{\text{HR}}) \in [0,1]^{H \times W \times 3} \tag{2.1}$$

using some downsampling operator

$$\text{d}_\alpha : [0,1]^{\alpha H \times \alpha W \times 3} \rightarrow [0,1]^{H \times W \times 3} \tag{2.2}$$

for a fixed scaling factor $\alpha > 1$, image height H, width W and 3 color channels. The task of SISR is to provide an approximate inverse $\text{f} \approx \text{d}^{-1}$ estimating I_{HR} from I_{LR}:

$$\text{f}(I_{\text{LR}}) = I_{\text{est}} \approx I_{\text{HR}}. \tag{2.3}$$

This problem is highly ill-posed as the downsampling operation d is non-injective and there exists a very large number of possible images I_{est} for which $\text{d}(I_{\text{est}}) = I_{\text{LR}}$ holds.

Recent learning approaches aim to approximate f via multi-layered neural networks by minimizing the Euclidean loss $||I_{\text{est}} - I_{\text{HR}}||_2^2$ between the current estimate and the ground truth image. While these models reach excellent results as measured by PSNR, the resulting images tend to look blurry and lack high frequency textures present in the original images. This is a direct effect of the high ambiguity in SISR: since downsampling removes high frequency information from the input image, no method can hope to reproduce all fine

details with pixel-wise accuracy. Therefore, even state-of-the-art models learn to produce the mean of all possible textures in those regions in order to minimize the Euclidean loss for the output image.

To illustrate this effect, we designed a simple toy example in Figure 2.2, where all high frequency information is lost by downsampling. The optimal solution with respect to the Euclidean loss is simply the average of all possible images while more advanced loss functions lead to more realistic, albeit not pixel-perfect reproductions.

2.4 Method

2.4.1 Architecture

Our network architecture in Table 2.1 (left) is inspired by Long *et al.* (2015) and Johnson *et al.* (2016) since feed-forward fully convolutional neural networks exhibit a number of useful properties for the task of SISR. The exclusive use of convolutional layers enables training of a single model for an input image of arbitrary size at a given scaling factor α while the feed-forward architecture results in an efficient model at inference time since the LR image only needs to be passed through the network once to get the result. The exclusive use of 3×3 filters is inspired by the VGG architecture (Simonyan and Zisserman, 2015) and allows for deeper models at a low number of parameters in the network.

As the LR input is smaller than the output image, it needs to be upsampled at some point to produce a high-resolution image estimate. It may seem natural to simply feed the bicubic interpolation of the LR image into the network (Dong *et al.*, 2014). However, this introduces redundancies to the input image and leads to a higher computational cost. For convolutional neural networks, Long *et al.* (2015) use convolution transpose layers[1] which upsample the feature activations inside the network. This circumvents the nuisance of having to feed a large image with added redundancies into the CNN and allows most computation to be done in the LR image space, resulting in a smaller network and larger receptive fields of the filters relative to the output image.

However, convolution transpose layers have been reported to produce checkerboard artifacts in the output, necessitating an additional regularization term in the output such as total variation (Rudin *et al.*, 1992). Odena *et al.* (2016) replace the convolution transpose layers with nearest-neighbor upsampling of the feature activations in the network followed by a single convolution layer. In our network architecture, this approach still produces checkerboard-artifacts for some specific loss functions, however we found that it obviates the need for an additional regularization term in our more complex models. To further reduce artifacts, we add a convolution layer after all upsampling blocks in the HR image space as this helps to avoid regular patterns in the output.

[1] Long *et al.* (2015) introduce them as *deconvolution* layers which may be misleading since no actual deconvolution is performed. Other names for convolution transpose layers include *upconvolution*, *fractionally strided convolution* or simply *backwards convolution*.

Output size	Layer
$H \times W \times 3$	Input I_{LR}
$H \times W \times 64$	Conv, ReLU
	Res: Conv, ReLU, Conv
	⋮
	Res: Conv, ReLU, Conv
$2H \times 2W \times 64$	2x NN upsampling
	Conv, ReLU
$4H \times 4W \times 64$	2x NN upsampling
	Conv, ReLU
	Conv, ReLU
$4H \times 4W \times 3$	Conv
	Residual image I_{res}
$4H \times 4W \times 3$	Output $I_{\text{est}} = I_{\text{bicubic}} + I_{\text{res}}$

Output size	Layer
$128 \times 128 \times 3$	Input I_{est} or I_{HR}
$128 \times 128 \times 32$	Conv, lReLU
$64 \times 64 \times 32$	Conv stride 2, lReLU
$64 \times 64 \times 64$	Conv, lReLU
$32 \times 32 \times 64$	Conv stride 2, lReLU
$32 \times 32 \times 128$	Conv, lReLU
$16 \times 16 \times 128$	Conv stride 2, lReLU
$16 \times 16 \times 256$	Conv, lReLU
$8 \times 8 \times 256$	Conv stride 2, lReLU
$8 \times 8 \times 512$	Conv, lReLU
$4 \times 4 \times 512$	Conv stride 2, lReLU
8192	Flatten
1024	Fc, lReLU
1	Fc, sigmoid
1	Estimated label

Table 2.1: Our generative fully convolutional network architecture for 4x super-resolution (left) and the discriminative network used for the adversarial loss (right). The generative network only learns the residual image between the bicubic interpolation of the input and the ground truth. We use 3×3 convolution kernels, 10 residual blocks in the generative network and we train on RGB images. The design of the discriminative network draws inspiration from VGG but uses leaky ReLU activations and strided convolutions instead of pooling layers.

Training deep networks, we found residual blocks (He *et al.*, 2016) to be beneficial for faster convergence compared to stacked convolution layers. A similarly motivated idea proposed by (Kim *et al.*, 2016a) is to learn only the residual image by adding the bicubic interpolation of the input to the model's output, so that it does not need to learn the identity function for I_{LR}. While the residual blocks that make up a main part of our network already only add residual information, we found that applying this idea helps stabilize training and reduce color shifts in the output during training.

2.4.2 Training and Loss Functions

In this section, we introduce the loss terms used to train our network. Various combinations of these losses and their effects on the results are discussed in Section 2.5.1. After introducing the classical pixel-wise Euclidean loss for training super-resolution networks, we continue to describe the perceptual loss and texture synthesis loss. Finally, we introduce the adversarial loss which leads to overall sharper images.

Bicubic ENet-E ENet-PAT Ground Truth

Figure 2.3: Our results on an image from ImageNet for 4x super-resolution. Despite reaching state-of-the-art results by PSNR, ENet-E produces an unnatural and blurry image while ENet-PAT reproduces faithful high-frequency information, resulting in a photorealistic image, at first glance almost indistinguishable from the ground truth image.

Pixel-wise loss in the image-space

As a baseline, we train our model with the pixel-wise MSE

$$\mathcal{L}_E = ||I_{\text{est}} - I_{\text{HR}}||_2^2, \quad \text{where} \quad ||I||_2^2 = \frac{1}{HW} \sum_{H,W} (I_{H,W})^2. \tag{2.4}$$

This simple loss is classically used to train super-resolution networks, penalizing variations in intensity for each color channel in RGB images.

Perceptual loss in feature space

Dosovitskiy and Brox (Dosovitskiy and Brox, 2016) as well as Johnson *et al.* (2016) propose a *perceptual similarity measure*. Rather than computing distances in image space, both I_{est} and I_{HR} are first mapped into a feature space by a differentiable function ϕ before computing their distance.

$$\mathcal{L}_P = ||\phi(I_{\text{est}}) - \phi(I_{\text{HR}})||_2^2 \tag{2.5}$$

This allows the model to generate outputs that may not match the ground truth image with pixel-wise accuracy but instead encourages the network to produce images that have similar feature representations.

For the feature map ϕ, we use a pre-trained implementation of the popular VGG-19 network (Simonyan and Zisserman, 2015; Machrisaa, 2016). It consists of stacked convolutions coupled with pooling layers to gradually decrease the spatial dimension of the image and to extract higher-level features in higher layers. To capture both low-level and high-level features, we use a combination of the second and fifth pooling layers and compute the MSE on their feature activations.

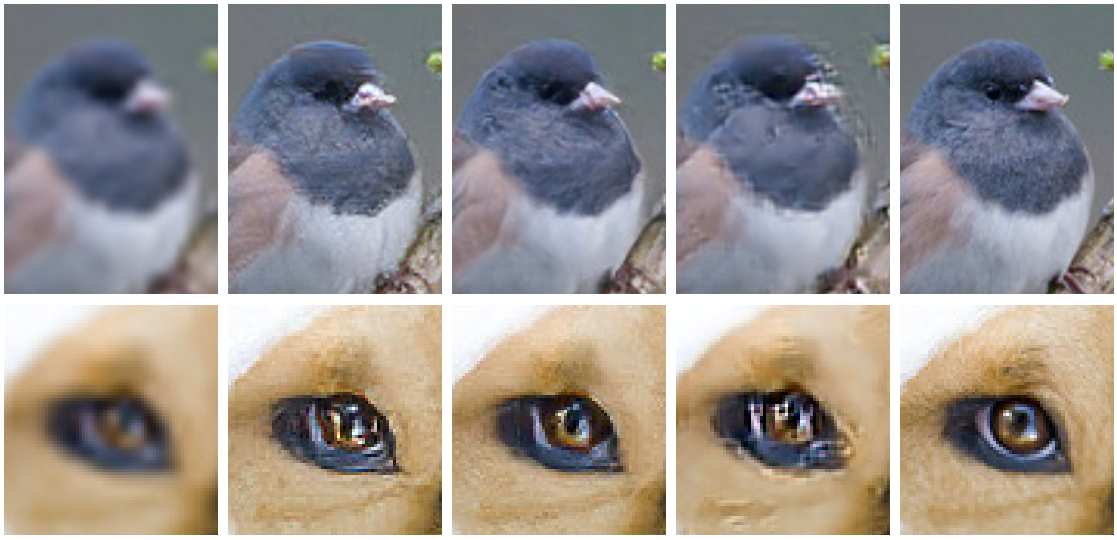


Figure 2.4: Comparing different patch sizes for the texture matching loss during training for ENet-PAT on images from ImageNet at 4x super-resolution. Computing the texture matching loss on small patches fails to capture textures properly (ENet-PAT-4) while matching textures on the whole image leads to unpleasant results since different texture statistics are averaged (ENet-PAT-128). We therefore use a patch size of 16 in all experiments (ENet-PAT-16, simplified to ENet-PAT in the following).

Texture matching loss

Gatys *et al.* (2015, 2016) demonstrate how convolutional neural networks can be used to create high quality textures. Given a target texture image, the output image is generated iteratively by matching statistics extracted from a pre-trained network to the target texture. As statistics, correlations between the feature activations $\phi(I) \in \mathbb{R}^{n \times m}$ at a given VGG layer with n features of length m are used:

$$\mathcal{L}_T = ||G(\phi(I_{\text{est}})) - G(\phi(I_{\text{HR}}))||_2^2, \tag{2.6}$$

with Gram matrix $G(F) = FF^T \in \mathbb{R}^{n \times n}$. As it is based on iterative optimization, this method is slow and only works if a target texture is provided at test time. Subsequent works train a feed-forward network that is able to synthesize a global texture (*e.g.*, a given painting style) onto other images (Johnson *et al.*, 2016; Ulyanov *et al.*, 2016), however a single network again only produces a single texture, and textures in all input images are replaced by the single style that the network has been trained for.

We propose using the style transfer loss for SISR: Instead of supplying our network with matching high-resolution textures during inference, we compute the texture loss

$\mathcal{L}_T$ patch-wise during training to enforce locally similar textures between I_{est} and I_{HR}. The network therefore learns to produce images that have the same local textures as the high-resolution images during training. While the task of generating arbitrary textures is more demanding than single-texture synthesis, the LR image and high-level contextual cues give our network more information to work with, enabling it to generate varying high resolution textures.

Empirically, we found a patch size of 16×16 pixels to result in the best balance between faithful texture generation and the overall perceptual quality of the images. Figure 2.4 shows ENet-PAT when trained using patches of size 4×4 pixels for the texture matching loss (ENet-PAT-4) and when it is calculated on larger patches of 128×128 pixels (ENet-PAT-128). Using smaller patches leads to artifacts in textured regions while calculating the texture matching loss on too large patches during training leads to artifacts throughout the entire image since the network is trained with texture statistics that are averaged over regions of varying textures, leading to unpleasant results.

Adversarial training

Adversarial training (Goodfellow *et al.*, 2014) is a recent technique that has proven to be a useful mechanism to produce realistically looking images. In the original setting, a generative network G is trained to learn a mapping from random vectors z to a data space of images x that is determined by the selected training dataset. Simultaneously, a discriminative network D is trained to distinguish between real images x from the dataset and generated samples $G(z)$. This approach leads to a minimax game in which the generator is trained to minimize

$$\mathcal{L}_A = -\log(D(G(z))) \tag{2.7}$$

while the discriminator minimizes

$$\mathcal{L}_D = -\log(D(x)) - \log(1 - D(G(z))). \tag{2.8}$$

In the SISR setting, G is our generative network as shown in Figure 2.1 (left), *i.e.*, the input to G is now an LR image I_{LR} instead of a noise vector z and its desired output is a suitable realistic high-resolution image I_{est}.

Following common practice (Radford *et al.*, 2016a), we apply leaky ReLU activations (Maas *et al.*, 2013) and use strided convolutions to gradually decrease the spatial dimensions of the image in the discriminative network as we found deeper architectures to result in images of higher quality. Figure 2.1 (right) shows the architecture of our discriminative adversarial network used for the loss term $\mathcal{L}_A$. We follow common design patterns and exclusively use convolutional layers with filters of size 3×3 pixels with varying stride lengths to reduce the spatial dimension of the input down to a size of 4×4 pixels where we append two fully connected layers along with a sigmoid activation at the output to

Network	Loss	Description	Loss	Weights
ENet-E	$\mathcal{L}_E$	Baseline with MSE	$\mathcal{L}_P$	$2 \cdot 10^{-1}$ pool_2
ENet-P	$\mathcal{L}_P$	Perceptual loss		$2 \cdot 10^{-2}$ pool_5
ENet-EA	$\mathcal{L}_E + \mathcal{L}_A$	ENet-E + adversarial	$\mathcal{L}_A$	1
ENet-PA	$\mathcal{L}_P + \mathcal{L}_A$	ENet-P + adversarial	$\mathcal{L}_T$	$3 \cdot 10^{-7}$ $\text{conv}_{1.1}$
ENet-EAT	$\mathcal{L}_E + \mathcal{L}_A + \mathcal{L}_T$	ENet-EA + texture loss		$1 \cdot 10^{-6}$ $\text{conv}_{2.1}$
ENet-PAT	$\mathcal{L}_P + \mathcal{L}_A + \mathcal{L}_T$	ENet-PA + texture loss		$1 \cdot 10^{-6}$ $\text{conv}_{3.1}$

Table 2.2: Overview of different combinations of loss functions for EnhanceNet (left) and weights for the different objectives (right). The weight for the adversarial loss is fixed to 1 in all models with the exception of ENet-PAT where it is set to 2.

produce a classification label between 0 and 1. Perhaps surprisingly, we found dropout not to be effective at preventing the discriminator from overpowering the generator. Instead, the following learning strategy yields better results and a more stable training: we keep track of the average performance of the discriminator on true and generated images within the previous training batch and only train the discriminator in the subsequent step if its performance on either of those two samples is below a threshold.

2.5 Evaluation

We first investigate the performance of our architecture trained with different combinations of the previously introduced loss functions in Section 2.5.1. After identifying the best performing models, we analyze the residual image that the network yields in Section 2.5.2 before presenting a comprehensive qualitative and quantitative evaluation of our approach in comparison to previous works in Sections 2.5.3–2.5.6. We finish the experimental section with a study of specialized training datasets in Section 2.5.7 and some notes on training details and model efficiency in Section 2.5.8.

2.5.1 Effect of Different Losses

We compare the performance of our network trained with the combinations of loss functions listed in Table 2.2. The results are shown in Figure 2.5 and Table 2.3.

ENet-E significantly sharpens details in the image compared to the bicubic interpolation. The perceptual loss in ENet-P yields slightly sharper results than ENet-E but it produces artifacts without adding new details in textured areas. Even though the perceptual loss is invariant under perceptually similar transformations, the network is given no incentive to produce realistic textures when trained with the perceptual loss alone.

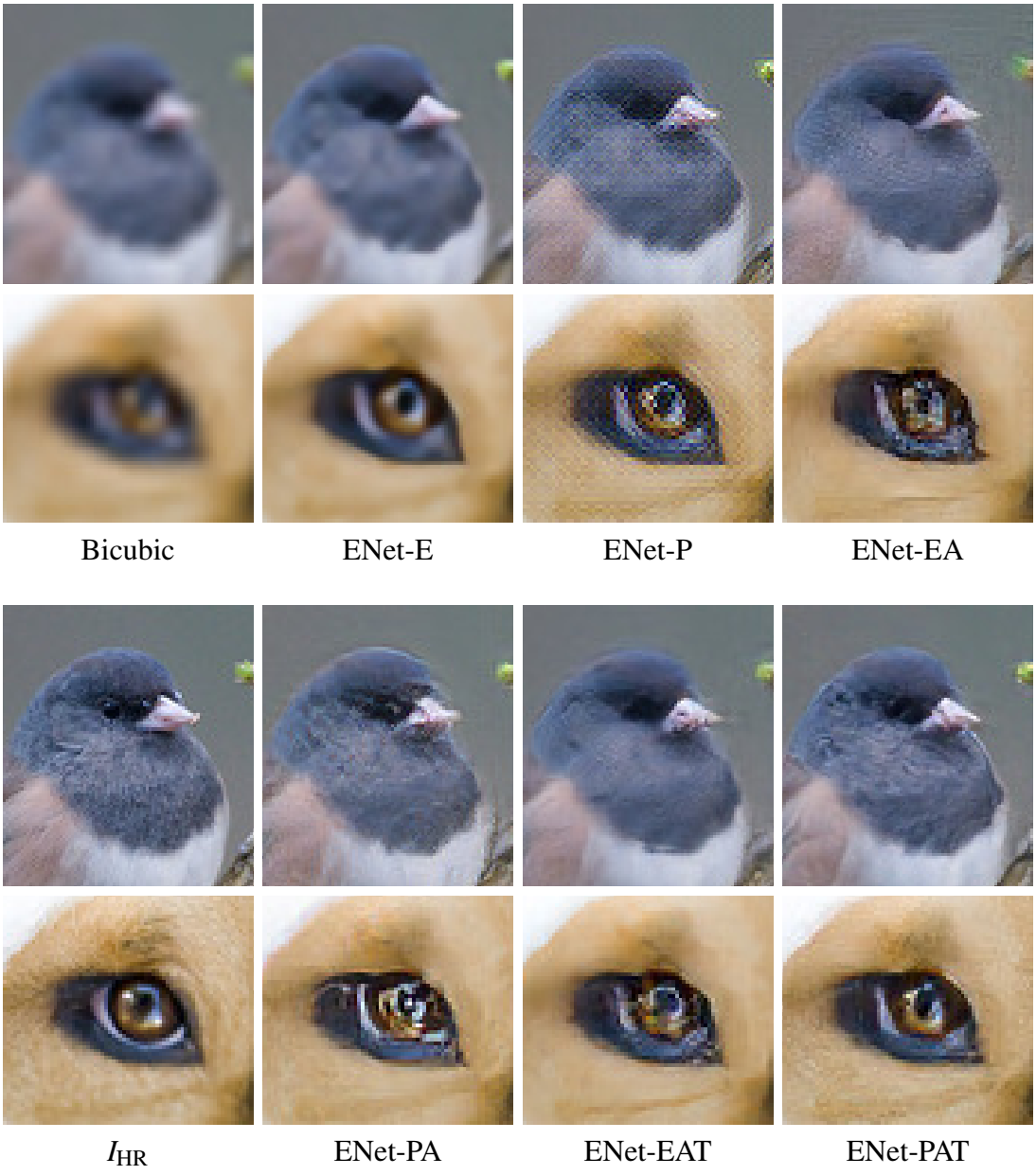


Figure 2.5: Comparing the results of our model trained with different losses at 4x super-resolution on images from ImageNet. ENet-P's result looks slightly sharper than ENet-E's, but it also produces unpleasing checkerboard artifacts. ENet-PA produces images that are significantly sharper but contain unnatural textures while we found that ENet-PAT generates more realistic textures, resulting in photorealistic images close to the original HR images. Replacing the perceptual loss in ENet-PA and ENet-PAT with the Euclidean loss results in images with sharp but jagged edges and overly smooth textures. Furthermore, these models are significantly harder to train since the Euclidean loss conflicts with the other loss terms.

Dataset	Bicubic	E	P	EA	PA	EAT	PAT
Set5	28.42	**31.74**	28.28	28.15	27.20	29.26	28.56
Set14	26.00	**28.42**	25.64	25.94	24.93	26.53	25.77
BSD100	25.96	**27.50**	24.73	25.71	24.19	25.97	24.93
Urban100	23.14	**25.66**	23.75	23.56	22.51	24.16	23.54

Table 2.3: PSNR for our architecture trained with different combinations of losses at 4x super resolution. ENet-E yields the highest PSNR values since it is trained towards minimizing the per-pixel distance to the ground truth. The models trained with the perceptual loss all yield lower PSNRs as it allows for deviations in pixel intensities from the ground truth. It is those outliers that significantly lower the PSNR scores. The texture loss increases the PSNR values by reducing the artifacts from the adversarial loss term. Best results shown in bold.

ENet-PA produces greatly sharper images by adding high frequency details to the output. However, the network sometimes produces unpleasing high-frequency noise to smooth regions and it seems to add high frequencies at random edges resulting in halos and sharpening artifacts in some cases. The texture loss helps ENet-PAT create locally meaningful textures and greatly reduces the artifacts. For some images, the results are almost indistinguishable from the ground truth even at a high magnification ratio of 4.

In general, we found training models with the adversarial and texture matching loss in conjunction with the Euclidean loss (in place of the perceptual loss) to be significantly less stable and the perceptual quality of the results oscillated heavily during training, *i.e.*, ENet-EA and ENet-EAT are harder to train than ENet-PA and ENet-PAT. This is because the adversarial and texture losses encourage the synthesis of high frequency information in the results, increasing the Euclidean distance to the ground truth images during training which leads to loss functions that counteract each other. The perceptual loss on the other hand is more tolerant to small-scale deviations due to pooling. We note that the texture matching loss in ENet-EAT leads to a more stable training than ENet-EA and slightly better results, though worse than ENet-PAT. This means that the texture matching loss not only helps create more realistic textures, but it also stabilizes the adversarial training.

Unsurprisingly, ENet-E yields the highest PSNR as it is optimized specifically for that measure. Although ENet-PAT produces perceptually more realistic images, the PSNR is much lower as the reconstructions are not pixel-accurate. SSIM, which has been found to correlate better with human perception (Yang *et al.*, 2014) also does not capture the perceptual quality of the results, so we provide alternative quantitative evaluations that agree better with human perception in Section 2.5.5 and Section 2.5.6.

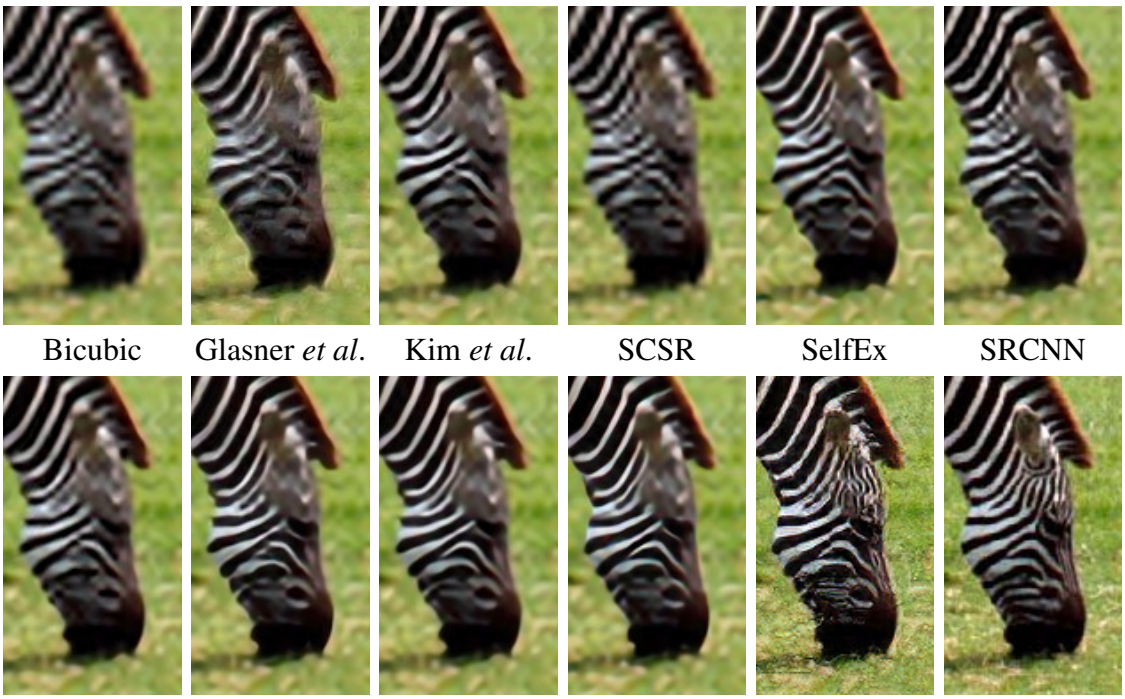


Figure 2.6: A comparison of previous methods with our results at 4x super-resolution on an image from Set14. Previous methods have continuously improved upon the restoration of sharper edges yielding higher PSNR's, a trend that ENet-E continues with slightly sharper edges and finer details (*e.g.*, area below the eye). With our texture-synthesizing approach, ENet-PAT is the only method that yields sharp lines and reproduces textures, resulting in the most realistic looking image. Furthermore, ENet-PAT not only sharpens but also produces high-frequency patterns missing completely in the LR image, *e.g.*, lines on the zebra's forehead or the grass texture, showing that the model is capable of detecting and generating patterns that lead to a realistic image.

2.5.2 Residual Learning

Our models only learn the residual image between the bicubic upsampled input image and the high resolution output which renders training more stable. Figure 2.12 on page 27 displays examples for residual images that our models estimate. ENet-E has learned to significantly increase the sharpness of the image and to remove aliasing effects in the bicubic interpolation (as seen in the aliasing effects in the residual image that cancel out with the aliasing in the bicubic interpolation). ENet-PAT additionally generates fine high-frequency textures in regions that should be textured while leaving smooth areas such as the sky and the red front areas of the house untouched.

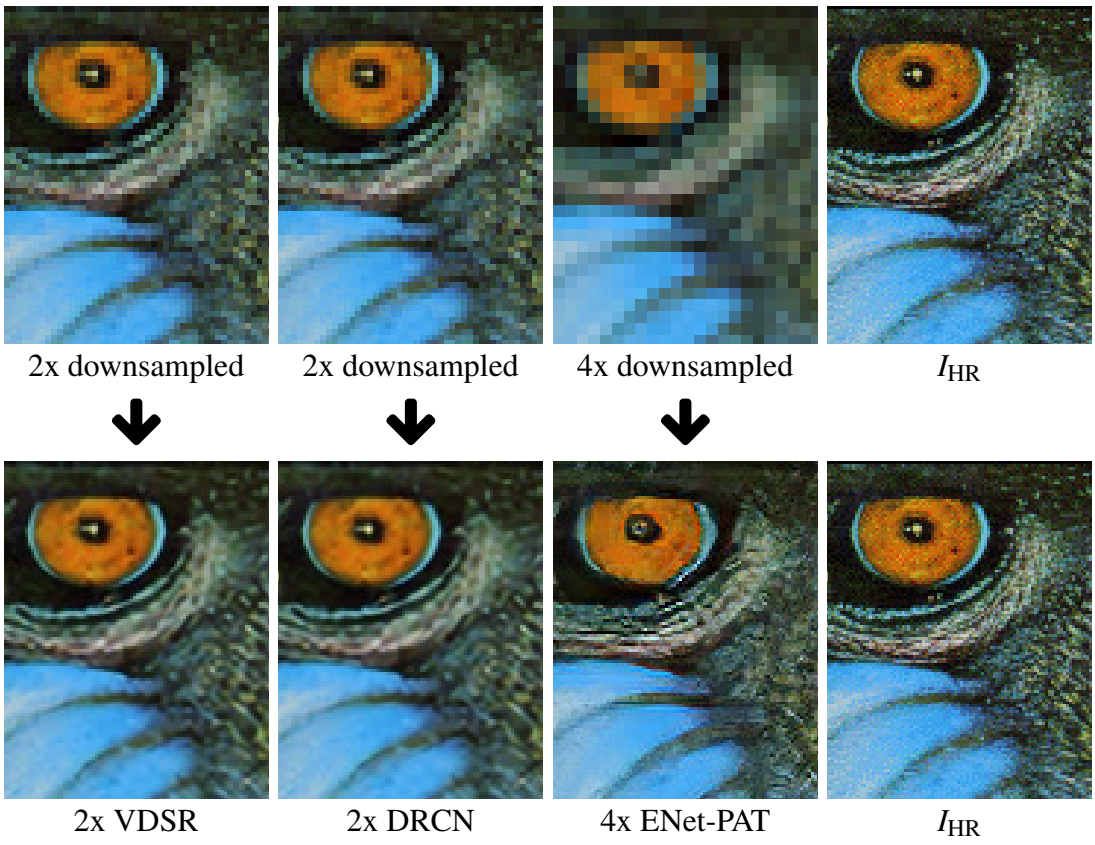


Figure 2.7: Comparing the previous state of the art by PSNR value at 2x super-resolution (75% of all pixels missing) with our model at 4x super-resolution (93.75% of all pixels missing). The top row shows the input to the models and the bottom row the results. Although our model has significantly less information to work with, it produces a sharper image with realistic textures.

2.5.3 Comparison with other Approaches

Figure 2.6 gives an overview of different approaches including the current state of the art by PSNR (Kim *et al.*, 2016a,b) on the zebra image from Set14 which is particularly well-suited for a visual comparison since it contains both smooth and sharp edges, textured regions as well as repeating patterns. Previous methods have gradually improved on edge reconstruction, but even the state-of-the-art model DRCN suffers from blur in regions where the LR image doesn't provide any high frequency information. While ENet-E reproduces slightly sharper edges, the results exhibit the same characteristics as previous approaches. ENet-PAT is the only model that produces significantly sharper images with realistic textures. It is interesting to see that ENet-PAT has learned to hallucinate detailed high-frequency patters as seen in the zebra's forehead.

Comparisons with Johnson *et al.* (2016), Bruna *et al.* (2016) and RAISR (Romano *et al.*, 2016) are shown in Figure 2.8.

Johnson *et al.* (2016) were the first to introduce the perceptual loss for use in super-resolution. Due to the fact that the perceptual loss is invariant to minor pixel-level differences and as a result of the pooling layers in the applied VGG architecture, the perceptual loss on its own produces heavy artifacts (*c.f.* ENet-P in Figure 2.5). They therefore apply an additional total variation regularizer to reduce the artifacts. As can be seen in Figure 2.8 (top), their result still produces checkerboard artifacts and the addition of the total variation regularization leads to blurry textures. In direct comparison, the result of ENet-PAT is significantly sharper with finer details and without artifacts.

Bruna *et al.* (2016) propose several models (scatter, fine-tuned, VGG) intended to replace MSE-optimized super-resolution in favor of more perceptually pleasing results. The comparison with ENet-PAT shows that we achieve sharper images with clear lines while the results of Bruna *et al.* (2016) contain more artifacts and jagged edges.

Since RAISR has been designed for speed rather than state-of-the-art image quality, it reaches a lower performance than previous methods (Perez-Pellitero *et al.*, 2016; Kim *et al.*, 2016a) so ENet-E yields visually sharper images even at this low scaling factor of 2x super-resolution. ENet-E already produces much sharper images than RAISR, though ENet-PAT is the only model to reconstruct sharp details and it is visually almost indistinguishable from the ground truth. Despite not being optimized for speed, EnhanceNet is even faster than RAISR at test-time: 9/18ms (EnhanceNet) vs. 17/30ms (RAISR) on average per image at 4x super-resolution on Set5/Set14, though EnhanceNet runs on a GPU while RAISR has been benchmarked on a 6-core CPU.

To demonstrate the significance of the jump in quality achieved by our method, we further compare the result of ENet-PAT at 4x super-resolution with the current state of the art models at 2x super-resolution in Figure 2.7. Although 4x super-resolution is a greatly more demanding task than 2x super-resolution, the results are comparable in quality. Small details that are lost completely in the 4x downsampled image are more accurate in VDSR and DRCN's outputs, but our model produces a plausible image with sharper textures at 4x super-resolution that even outperforms the current state of the art at 2x super-resolution in sharpness, *e.g.*, the area below the eyes is sharper in ENet-PAT's result and looks very similar to the ground truth.

2.5.4 Quantitative results by PSNR and SSIM

Table 2.4 summarizes the PSNR and SSIM values of our model in comparison to other approaches including the previous state of the art on various popular SISR benchmarks. ENet-E beats all prior works by a wide margin on 4x super-resolution across all datasets and on almost all datasets at 2x super-resolution.

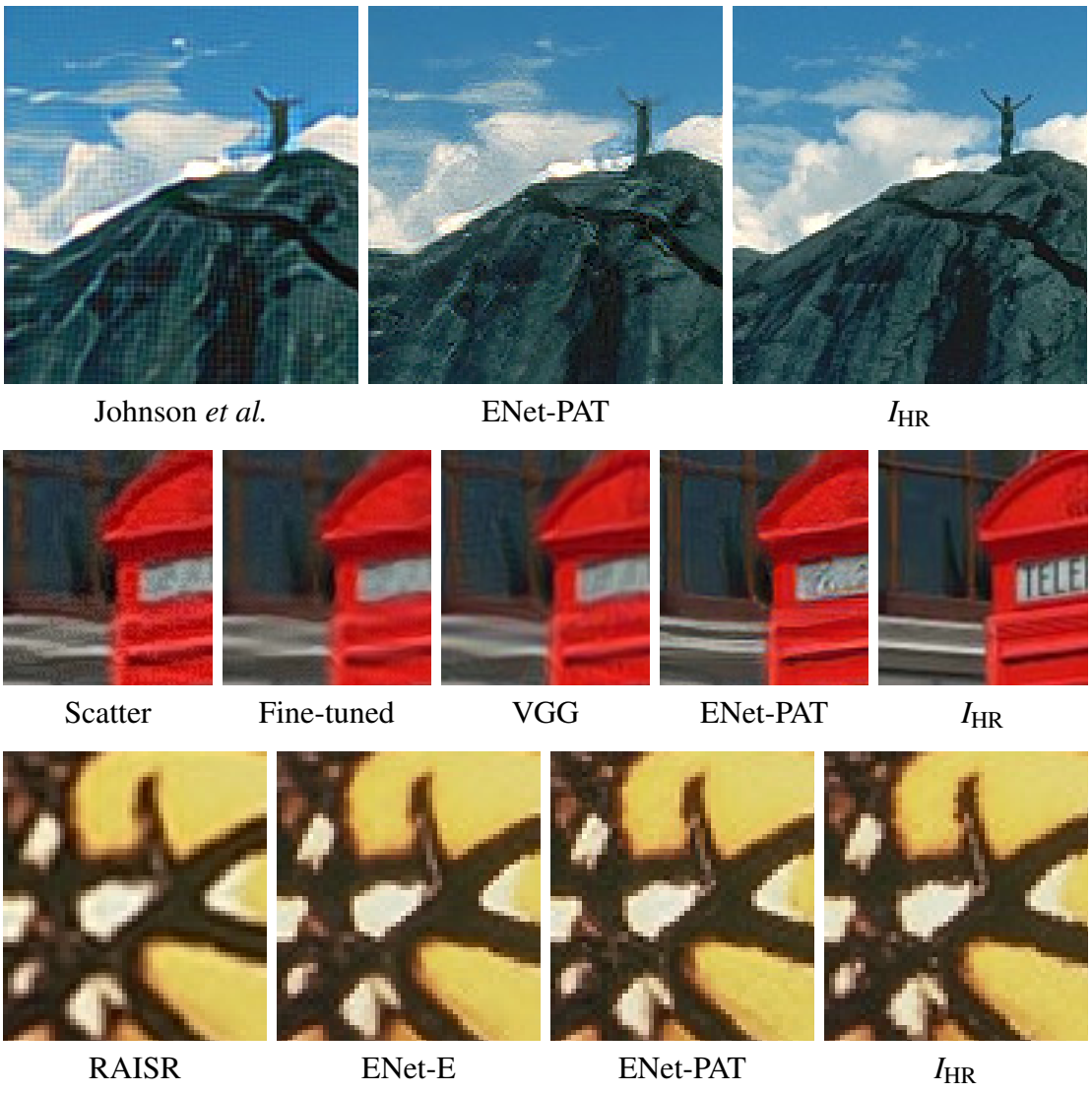


Figure 2.8: **(Top)** Comparing our model with the perceptual method by Johnson *et al.* (2016) on an image from BSD100 4x super-resolution. ENet-PAT's result looks more natural and does not contain checkerboard artifacts despite the lack of an additional regularization term. **(Middle)** Comparison with the 3 models Scatter, Fine-tuned, and VGG from Bruna *et al.* (2016) at 4x super-resolution, and a comparison with the efficient method RAISR (Romano *et al.*, 2016) at 2x super-resolution. ENet-PAT produces images with more contrast and sharper edges that are more faithful to the ground truth (see *e.g.* the edge at the bottom). **(Bottom)** Comparing our model with Romano *et al.* (2016) at 2x super-resolution on the butterfly image of Set5. Despite the low scaling factor, image quality gradually increases between RAISR, ENet-E and ENet-PAT, the last of which is not only sharper but also recreates small details better, *e.g.*, the vertical white line in the middle of the picture is fully reconstructed only in ENet-PAT's result.

2x PSNR	Bicubic	RFL	A+	SelfEx	SRCNN	PSyCo	DRCN	VDSR	ENet-E
Set5	33.66	36.54	30.14	36.49	36.66	36.88	**37.63**	37.53	37.32
Set14	30.24	32.26	27.24	32.22	32.42	32.55	33.04	33.03	**33.25**
BSD100	29.56	31.16	26.75	31.18	31.36	31.39	31.85	31.90	**31.95**
Urban100	26.88	29.11	24.19	29.54	29.50	29.64	30.75	30.76	**31.21**

2x SSIM	Bicubic	RFL	A+	SelfEx	SRCNN	PSyCo	DRCN	VDSR	ENet-E
Set5	0.930	0.954	0.954	0.954	0.954	0.956	**0.959**	**0.959**	0.958
Set14	0.869	0.904	0.906	0.903	0.906	0.898	0.912	0.912	**0.915**
BSD100	0.843	0.884	0.886	0.886	0.888	0.890	0.894	0.896	**0.898**
Urban100	0.840	0.871	0.894	0.895	0.895	0.900	0.913	0.914	**0.919**

4x PSNR	Bicubic	RFL	A+	SelfEx	SRCNN	PSyCo	DRCN	VDSR	ENet-E
Set5	28.42	30.14	30.28	30.31	30.48	30.62	31.53	31.35	**31.74**
Set14	26.00	27.24	27.32	27.40	27.49	27.57	28.02	28.01	**28.42**
BSD100	25.96	26.75	26.82	26.84	26.90	26.98	27.23	27.29	**27.50**
Urban100	23.14	24.19	24.32	24.79	24.52	24.62	25.14	25.18	**25.66**

4x SSIM	Bicubic	RFL	A+	SelfEx	SRCNN	PSyCo	DRCN	VDSR	ENet-E
Set5	0.810	0.855	0.860	0.862	0.863	0.868	0.885	0.884	**0.887**
Set14	0.703	0.745	0.749	0.752	0.750	0.753	0.867	0.767	**0.777**
BSD100	0.668	0.705	0.709	0.711	0.710	0.716	0.723	0.725	**0.733**
Urban100	0.658	0.710	0.718	0.737	0.722	0.732	0.751	0.752	**0.770**

Table 2.4: PSNR and SSIM for different methods at 2x and 4x super-resolution. We compare against RFL (Schulter *et al.*, 2015), A+ (Timofte *et al.*, 2014), SelfEx (Huang *et al.*, 2015a), SRCNN (Dong *et al.*, 2014), PSyCo (Perez-Pellitero *et al.*, 2016), DRCN (Kim *et al.*, 2016b), and VDSR (Kim *et al.*, 2016a). The progress over the years can be clearly seen here as more recent methods achieve progressively higher PSNR and SSIM scores. ENet-E follows this trend as it achieves state-of-the-art results on almost all datasets across both scales. As shown in Table 2.3, ENet-E is the best performing model among all other combinations of loss functions since it is the only model that exclusively optimizes for the Euclidean loss (other models not shown in this table). Best performance shown in bold.

2.5.5 Object recognition performance

It is known that super-resolution algorithms can be used as a preprocessing step to improve the performance of other image-related tasks such as face recognition (Fookes *et al.*, 2012). We propose to use the performance of state-of-the-art object recognition models as a metric to evaluate image reconstruction algorithms, especially for models whose performance is not captured well by PSNR and SSIM. For evaluation, any pre-trained object recognition model M and labeled set of images may be used. The image restoration models to be evaluated are applied on a degraded version of the dataset and the reconstructed images are fed into M. The hypothesis is that the performance of powerful object recognition models shows a meaningful correlation with the human perception of image quality that may complement pixel-based benchmarks.

Similar indirect metrics have been applied in previous works, *e.g.*, optical character recognition performance has been utilized to compare the quality of text deblurring algorithms (Hradiš *et al.*, 2015; Xiao *et al.*, 2016) and face-detection performance has been used for the evaluation of super-resolution algorithms (Lin *et al.*, 2007). The performance of object recognition models has been used for the indirect evaluation of image colorization (Zhang *et al.*, 2016), where black and white images were colorized to improve object detection rates. Namboodiri *et al.* (2011) apply a metric similar to ours to evaluate SISR algorithms and found it to be a better metric than PSNR or SSIM for evaluating the perceptual quality of super-resolved images.

For our comparison, we use ResNet-50 (Dahl, 2016; He *et al.*, 2016) as this class of models has achieved state-of-the-art performance by winning the 2015 Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky *et al.*, 2015). For the evaluation, we use the first 1000 images in the ILSVRC 2016 CLS-LOC validation dataset[2] where each image has exactly one out of 1000 labels. The original images are scaled to 224×224 for the baseline and downsampled to 56×56 for a scaling factor of 4. We report the mean top-1 and top-5 errors as well as the mean confidence that ResNet reports on correct classifications. The results are shown in Table 2.5. In our comparison, some of the results roughly coincide with the PSNR scores, with bicubic interpolation resulting in the worst performance followed by DRCN (Kim *et al.*, 2016b) and PSyCo (Perez-Pellitero *et al.*, 2016) which yield visually comparable images and hence similar scores as our ENet-E network. However, our models ENet-EA, ENet-PA and ENet-PAT produce images of higher perceptual quality which is reflected in higher classification scores despite their low PSNR scores. This indicates that the object recognition benchmark matches human perception better than PSNR does. The high scores of ENet-PAT are not a result of overfitting due to being trained with VGG, since even ENet-EA (which is not trained with VGG) gains higher scores than *e.g.* ENet-E, which has the highest PSNR but lower scores under this metric.

[2] We use the validation dataset since the annotations for the test dataset are not released. However, even a potential bias of the ResNet-model would not invalidate the results, since higher scores only imply that the upscaled images are closer to the originals under the proposed metric.

Evaluation	Bicubic	DRCN	PSyCo	ENet-E	ENet-EA	ENet-PA	ENet-PAT	Baseline
Top-1 error	0.506	0.477	0.454	0.449	0.407	0.429	**0.399**	0.260
Top-5 error	0.266	0.242	0.224	0.214	0.185	0.199	**0.171**	0.072
Confidence	0.754	0.727	0.728	0.754	0.760	0.783	**0.797**	0.882

Table 2.5: ResNet object recognition performance and reported confidence on pictures from the ImageNet dataset downsampled to 56×56 before being upscaled by a factor of 4 using different algorithms. The baseline shows ResNet's performance on the original 224×224 sized images. Compared to PSNR, the scores correlate better with the human perception of image quality: ENet-E achieves only slightly higher scores than DRCN or PSyCo since all these models minimize pixel-wise MSE. On the other hand, ENet-PAT achieves higher scores as it produces sharper images and more realistic textures. The good results of ENet-EA which is trained without VGG indicate that the high scores of ENet-PAT are not solely due to being trained with VGG, but likely a result of sharper images. Best results shown in bold.

While we observe that the object recognition performance roughly coincides with the human perception of image quality in this benchmark for super-resolution, we leave a more detailed analysis of this evaluation metric on other image restoration problems to future work.

2.5.6 Evaluation of perceptual quality

To further validate the perceptual quality of our results, we conducted a user study on the ImageNet dataset from the previous section. As a representative for models that minimize the Euclidean loss, we compare ENet-E as the new state of the art in PSNR performance with the images generated by ENet-PAT which have a PSNR comparable to images upsampled with bicubic interpolation. The subjects were shown the ground truth image along with the super-resolution results of both ENet-E and ENet-PAT at 4x super-resolution side-by-side, and were asked to select the image that looks more similar to the ground truth. Figure 2.13 on page 28 shows a screenshot of the applied survey. In 49 survey responses for a total of 843 votes, subjects selected the image produced by ENet-PAT 91.0% of the time, underlining the perceptual quality of our results.

2.5.7 Specialized Training Datasets

Figure 2.9 shows an example for an image where the majority of subjects in our survey preferred ENet-E's result over the image produced by ENet-PAT. In general, ENet-PAT trained on MSCOCO struggles to reproduce realistically looking faces at high scaling factors and while the overall image is significantly sharper than the result of ENet-E,

Bicubic ENet-E ENet-PAT I_{HR}

Figure 2.9: Failure case for ENet-PAT on an image from ImageNet at 4x super-resolution. While producing an overall sharper image than ENet-E, ENet-PAT fails to reproduce a realistically looking face, leading to a perceptually implausible result. An approach to mitigate this weakness is shown in Section 2.5.7 and Figure 2.11.

the human perception is highly sensitive to small changes in the appearance of human faces which is why many subjects preferred the blurry result of ENet-E in those cases. To demonstrate that this is not a limitation of our model, we train ENet-PAT with identical hyperparameters on the CelebA dataset (Liu *et al.*, 2015a) (ENet-PAT-F) and compare the results with ENet-PAT trained on MSCOCO as before. The results are shown in Figure 2.11 on page 26. Trained on CelebA, ENet-PAT-F shows significantly better performance than ENet-PAT trained on MSCOCO which does not contain many faces.

2.5.8 Training Details and Inference Speed

For training, we use all color images in MSCOCO (Lin *et al.*, 2014) that have at least 384 pixels on the short side resulting in roughly 200k images. All images are cropped centrally to a square and then downsampled to 256×256 to reduce noise and JPEG artifacts. During training, we fix the size of the input I_{LR} to 32×32. As the scale of objects in the MSCOCO dataset is too small when downsampled to such a small size, we downsample the 256×256 images by α and then crop these to patches of size 32×32. After training the model for any given scaling factor α, the input to the fully convolutional network at test time can be an image of arbitrary dimensions $H \times W$ which is then upscaled to $(\alpha H) \times (\alpha W)$.

The model has been implemented in TensorFlow r0.10 (Abadi et. al., 2015). For all weights, we apply Xavier initialization (Glorot and Bengio, 2010) and we train using the Adam optimizer (Kingma and Ba, 2015) with a fixed learning rate of 10^{-4}. We found common convolutional layers stacked with ReLU's to yield comparable results, but training converges faster with the residual architecture. All models were trained only once and used for all results throughout the manuscript, no fine-tuning was done for any specific dataset or image. Nonetheless, we believe that a choice of specialized training datasets for specific types of images can greatly increase the perceptual quality of the produced textures (*c.f.* Section 2.5.7).

Bicubic ENet-PAT I_{HR}

Figure 2.10: Failure case on an image from BSD100. ENet-PAT has learned to continue high-frequency patterns. While that works out extremely well in most cases (*c.f.* zebra's forehead in Figure 2.6), the model fails in this notable case as I_{HR} is smooth in that region.

We trained all models for a maximum of 24 hours on an Nvidia K40 GPU, though convergence rates depend on the applied combination of loss functions. Although not optimized for efficiency, our network is compact and quite fast at test time. The final trained model is only 3.1MB in size and processes images in 9ms (Set5), 18ms (Set14), 12ms (BSD100) and 59ms (Urban100) on average per image at 4x super-resolution.

2.6 Summary

We have proposed an architecture that is capable of producing state-of-the-art results by both quantitative and qualitative measures by training with a Euclidean loss or a novel combination of adversarial training, perceptual losses and a newly proposed texture transfer loss for super-resolution. Once trained, the model interpolates full color images in a single forward-pass at competitive speeds.

As SISR is a heavily ill-posed problem, some limitations remain. While images produced by ENet-PAT look realistic, they do not match the ground truth images on a pixel-wise basis. Furthermore, the adversarial training sometimes produces artifacts in the output which are greatly reduced but not fully eliminated with the addition of the texture loss. We noted an interesting failure on an image in the BSD100 dataset that is shown in Figure 2.10, where the model continues a pattern visible in the LR image onto smooth areas. This is a result of the model learning to hallucinate textures that occur frequently between pairs of LR and HR images such as repeating stripes that fade in the LR image as they increasingly shrink in size.

While the model is already competitive in terms of its runtime, future work may decrease the depth of the network and apply shrinking methods to speed up the model to real-time performance on high-resolution data: adding a term for temporal consistency could then enable the model to be used for video super-resolution. An implementation along with a trained model of ENet-PAT is available at github.com/msmsajjadi/EnhanceNet-Code.

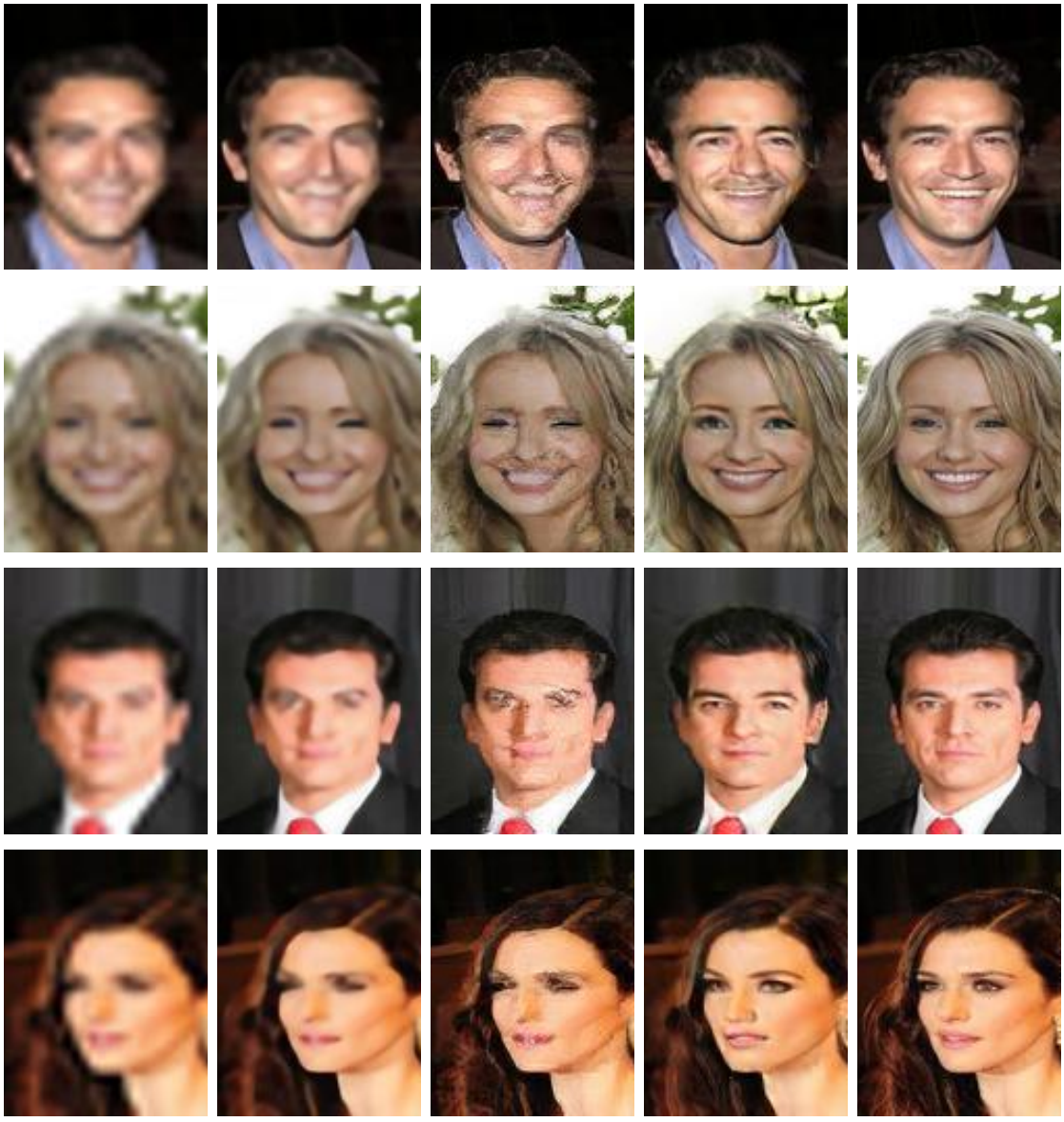
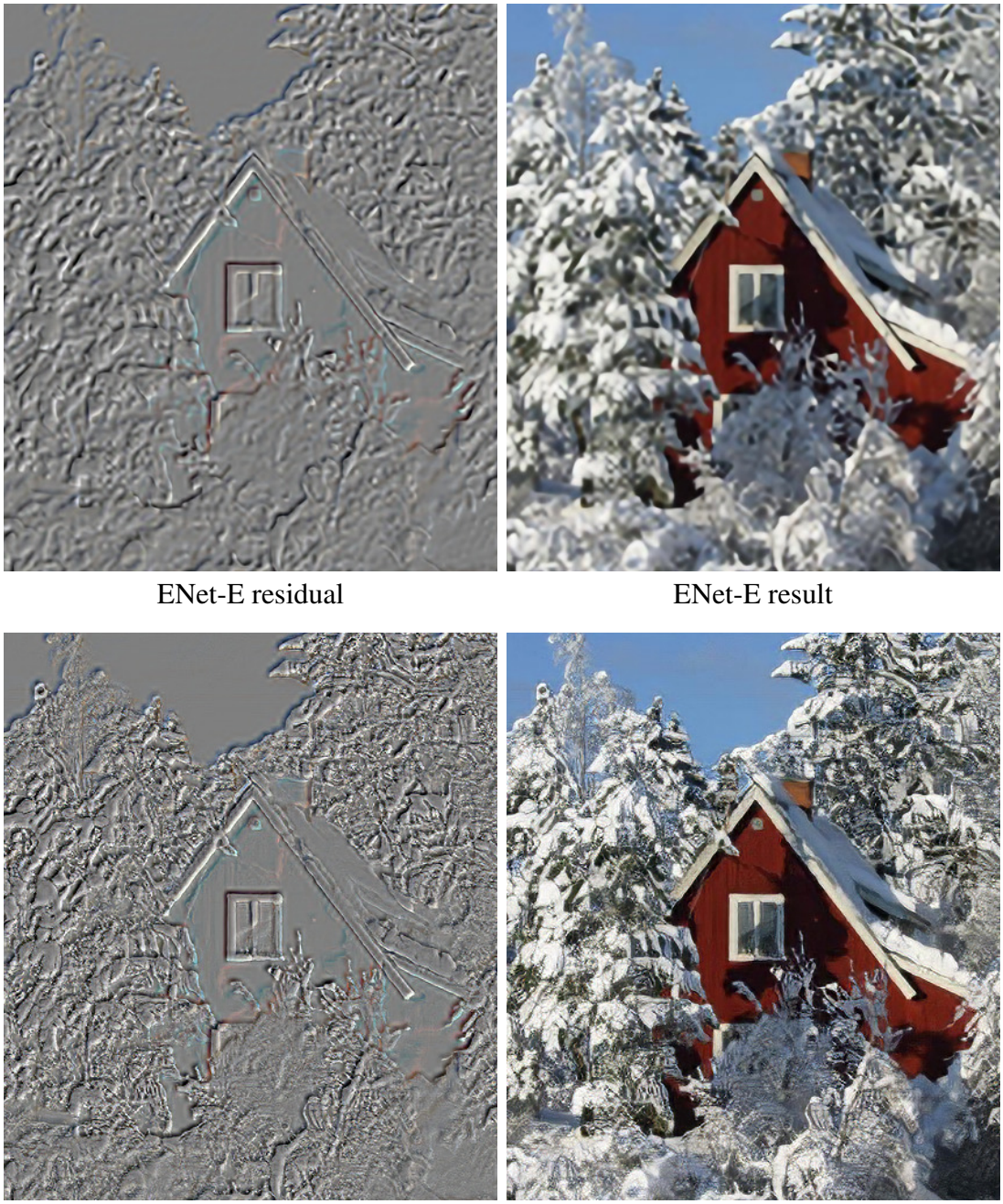


Figure 2.11: Comparing our models on images of faces at 4x super resolution. ENet-PAT trained on the MSCOCO dataset produces artifacts since its training dataset did not contain many high-resolution images of faces. When trained specifically on a dataset of faces (ENet-PAT-F), the same network produces realistic very realistic images, though the results look different from the actual ground truth images (similar to the results in Yu and Porikli (2016)). Note that we did not fine-tune the parameters of the losses for this specific task so even better results are likely achievable.



ENet-E residual | ENet-E result

ENet-PAT residual | ENet-PAT result

Figure 2.12: A visualization of the residual image that the network produces at 4x super-resolution. While ENet-E significantly sharpens edges and is able to remove aliasing from the bicubic interpolation, ENet-PAT produces additional textures yielding a sharp, realistic result. Image taken from the SunHays80 dataset (Sun and Hays, 2012).

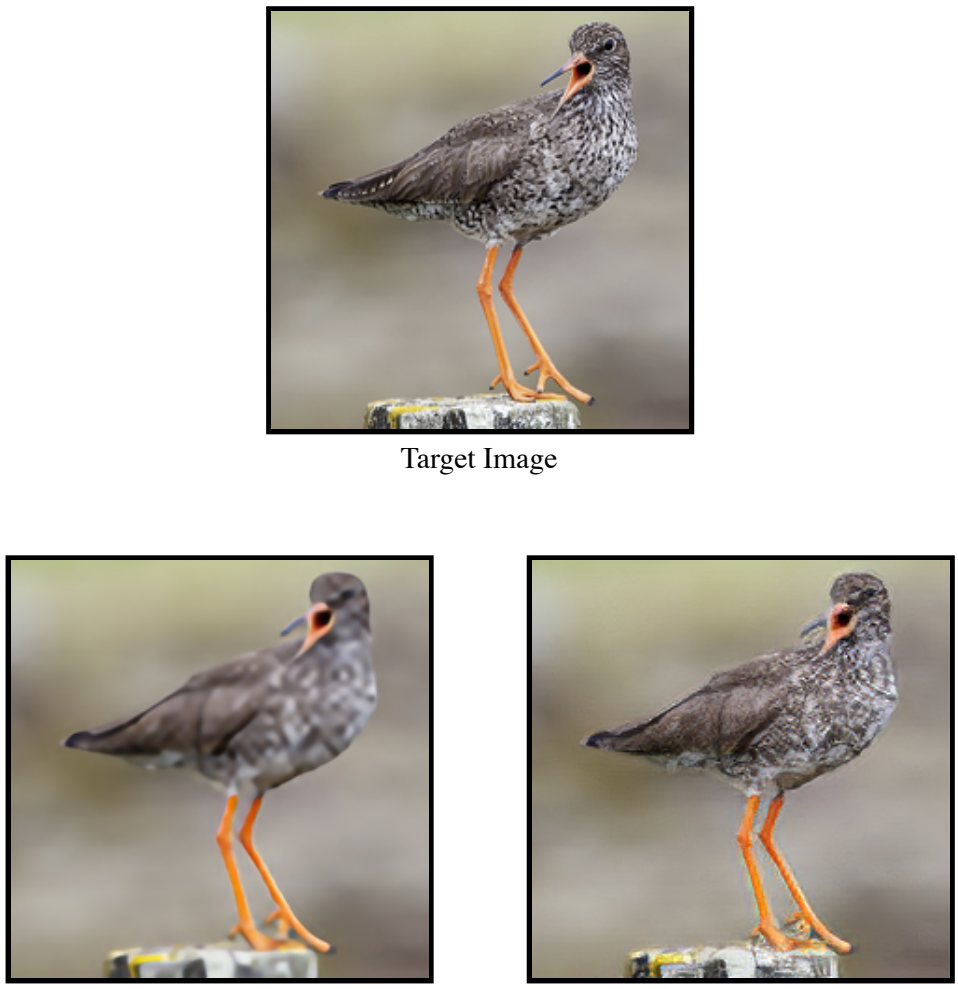


Figure 2.13: Example screenshot of our survey for perceptual image quality. Subjects were shown a target image above and were asked to select the image on the bottom that looks more similar to the target image. Each subject was shown up to 30 images. In 49 survey responses for a total of 843 votes, subjects selected the image produced by ENet-PAT 91.0%, underlining its higher perceptual quality compared to the state of the art by PSNR, ENet-E.

Chapter 3

Frame-Recurrent Video Super-Resolution

Recent advances in video super-resolution have shown that convolutional neural networks combined with motion compensation are able to merge information from multiple low-resolution (LR) frames to generate high-quality images. Current state-of-the-art methods process a batch of LR frames to generate a single high-resolution (HR) frame and run this scheme in a sliding window fashion over the entire video, effectively treating the problem as a large number of separate multi-frame super-resolution tasks. This approach has two main weaknesses: 1) Each input frame is processed and warped multiple times, increasing the computational cost, and 2) each output frame is estimated independently conditioned on the input frames, limiting the system's ability to produce temporally consistent results.

In this work, we propose an end-to-end trainable frame-recurrent video super-resolution framework that uses the previously inferred HR estimate to super-resolve the subsequent frame. This naturally encourages temporally consistent results and reduces the computational cost by warping only one image in each step. Furthermore, due to its recurrent nature, the proposed method has the ability to assimilate a large number of previous frames without increased computational demands. Extensive evaluations and comparisons with previous methods validate the strengths of our approach and demonstrate that the proposed framework is able to significantly outperform the current state of the art.

3.1 Introduction

Super-resolution is a classic problem in image processing that addresses the question of how to reconstruct a high-resolution (HR) image from its downscaled low-resolution (LR) version. With the rise of deep learning, super-resolution has received significant attention from the research community over the past few years (Dong *et al.*, 2014; Kim *et al.*, 2016a; Shi *et al.*, 2016a; Kappeler *et al.*, 2016; Tao *et al.*, 2017; Liu *et al.*, 2017; Caballero *et al.*, 2017; Sajjadi *et al.*, 2017; Ledig *et al.*, 2017). While high-frequency details need to be reconstructed exclusively from spatial statistics in the case of single image super-resolution, temporal relationships in the input can be exploited to improve reconstruction

LR input frame | Our result | HR frame

Figure 3.1: Side-by-side comparison of LR input frame, our FRVSR result, and HR ground truth for 4x upsampling. Thanks to its frame-recurrent architecture, our model reconstructs fine details that are missing from single input frames.

for video super-resolution. It is therefore imperative to combine the information from as many LR frames as possible to reach the best video super-resolution results.

The latest state-of-the-art video super-resolution methods approach the problem by combining a batch of LR frames to estimate a single HR frame, effectively dividing the task of video super-resolution into a large number of separate multi-frame super-resolution subtasks (Caballero *et al.*, 2017; Tao *et al.*, 2017; Liu *et al.*, 2017; Makansi *et al.*, 2017). However, this approach is computationally expensive since each input frame needs to be processed several times. Furthermore, generating each output frame separately reduces the system's ability to produce temporally consistent frames, resulting in unpleasing flickering artifacts.

In this work, we propose an end-to-end trainable frame-recurrent video super-resolution (FRVSR) framework to address the above issues. Instead of estimating each video frame separately, we use a recurrent approach that passes the previously estimated HR frame as an input for the following iteration. Using this recurrent architecture has several benefits. Each input frame needs to be processed only once which reduces the computational cost. Furthermore, information from past frames can be propagated to several later frames via the HR estimate that is recurrently passed through time. Passing the previous HR estimate directly to the next step helps the model to recreate fine details and produce temporally consistent videos in a more efficient way.

To analyze the performance of the proposed framework, we compare it with strong single image and video super-resolution baselines using identical neural networks as building blocks. Our extensive set of experiments provides insights into how the performance of FRVSR varies with the number of recurrent steps used during training, the size of the network, and the amount of noise, aliasing or compression artifacts present in the LR input. The proposed approach clearly outperforms the baselines under various settings both in terms of quality and efficiency. Finally, we also compare FRVSR with several existing video super-resolution approaches and show that it significantly outperforms the current state of the art on a standard benchmark dataset.

Our contributions

• We propose a recurrent framework that uses the HR estimate of the previous frame for generating the subsequent frame, leading to an efficient model that produces temporally consistent results.

• Unlike existing approaches, the proposed framework can propagate information over a large temporal range without increasing computations.

• Our system is end-to-end trainable and does not require any pre-training stages.

• We perform an extensive set of experiments to analyze the proposed framework and relevant baselines under various different settings.

• We show that the proposed framework significantly outperforms the current state of the art in video super-resolution both qualitatively and quantitatively.

3.2 Video Super-Resolution

Let $I_t^{\text{LR}} \in [0,1]^{H \times W \times 3}$ denote the t-th LR video frame obtained by downsampling the original HR video frame $I_t^{\text{HR}} \in [0,1]^{sH \times sW \times 3}$ by scale factor s. Given a set of consecutive LR video frames, the goal of video super-resolution is to generate HR estimates I_t^{est} that approximate the original HR frames I_t^{HR} under some metric.

3.2.1 Related Work

Super-resolution is a classic ill-posed inverse problem with approaches ranging from simple interpolation methods such as Bilinear, Bicubic and Lanczos (Duchon, 1979) to example-based super-resolution (Freeman *et al.*, 2002; Freedman and Fattal, 2011; Yang *et al.*, 2013; Timofte *et al.*, 2016), dictionary learning (Yang *et al.*, 2012; Perez-Pellitero *et al.*, 2016), and self-similarity approaches (Huang *et al.*, 2015a; Yang *et al.*, 2010a). We refer the reader to Milanfar (2010) and Nasrollahi and Moeslund (2014) for extensive overviews of prior art up to recent years.

The recent progress in deep learning, especially in convolutional neural networks, has shaken up the field of super-resolution. After Dong *et al.* (2014) reached state-of-the-art results with shallow convolutional neural networks, many others followed up with deeper network architectures, advancing the field tremendously (Shi *et al.*, 2016a; Kim *et al.*, 2016b,a; Dong *et al.*, 2016; Tai *et al.*, 2017; Lai *et al.*, 2017). Parallel efforts have studied alternative loss functions for more visually pleasing reconstructions (Ledig *et al.*, 2017; Sajjadi *et al.*, 2017). Agustsson and Timofte (2017) provide a recent survey on the current state of the art in single image super-resolution.

Video and multi-frame super-resolution approaches combine information from multiple LR frames to reconstruct details that are missing in individual frames which can lead to higher quality results. Classical video and multi-frame super-resolution methods are generally formulated as optimization problems that are computationally very expensive to solve (Farsiu *et al.*, 2004; Takeda *et al.*, 2009; Belekos *et al.*, 2010; Liu and Sun, 2011).

Most of the existing deep learning-based video super-resolution methods divide the task of video super-resolution into multiple separate sub-tasks, each of which generates a single HR output frame from multiple LR input frames. Kappeler *et al.* (2016) warp video frames I_{t-1}^{LR} and I_{t+1}^{LR} onto the frame I_t^{LR} using the optical flow method of Drulea and Nedevschi (2011), concatenate the three frames and pass them through a convolutional neural network that produces the output frame I_t^{est}. Caballero *et al.* (2017) follow the same approach but replace the optical flow model with a trainable motion compensation network. Makansi *et al.* (2017) follow an approach similar to Caballero *et al.* (2017) but combine warping and mapping to HR space into a single step. Tao *et al.* (2017) rely on a batch of up to 7 input LR frames to estimate a single HR frame. After computing the motion from neighboring input frames to I_t^{LR}, they map the frames onto high-resolution grids. In a final step, they run an encoder-decoder style network with a Conv-LSTM in the core yielding I_t^{est}. Liu *et al.* (2017) process up to 5 LR frames using different numbers of input frames (I_t^{LR}), $(I_{t-1}^{\mathrm{LR}}, I_t^{\mathrm{LR}}, I_{t+1}^{\mathrm{LR}})$, and $(I_{t-2}^{\mathrm{LR}}, \ldots, I_{t+2}^{\mathrm{LR}})$ simultaneously to produce separate HR estimates that are aggregated in a final step with dynamic weights to produce a single output I_t^{est}.

While a number of the above mentioned methods are end-to-end trainable, the authors often note that they first pre-train each component before fine-tuning the system as a whole in a final step (Caballero *et al.*, 2017; Liu *et al.*, 2017; Tao *et al.*, 2017).

Huang *et al.* (2015b) use a bidirectional recurrent architecture for video super-resolution with shallow networks but do not use any explicit motion compensation in their model. Recurrent architectures have also been used for other tasks such as video deblurring (Kim *et al.*, 2017) and stylization (Chen *et al.*, 2017a; Gupta *et al.*, 2017). While Kim *et al.* (2017) and Chen *et al.* (2017a) pass on a feature representation to the next step, Gupta *et al.* (2017) pass the previous output frame to the next step to produce temporally consistent stylized videos in concurrent work. A recurrent approach for video super-resolution was proposed by Farsiu *et al.* (2006) more than a decade ago with motivations similar to ours. However, this approach uses an approximation of the Kalman filter for frame estimation and is constrained to translational motion.

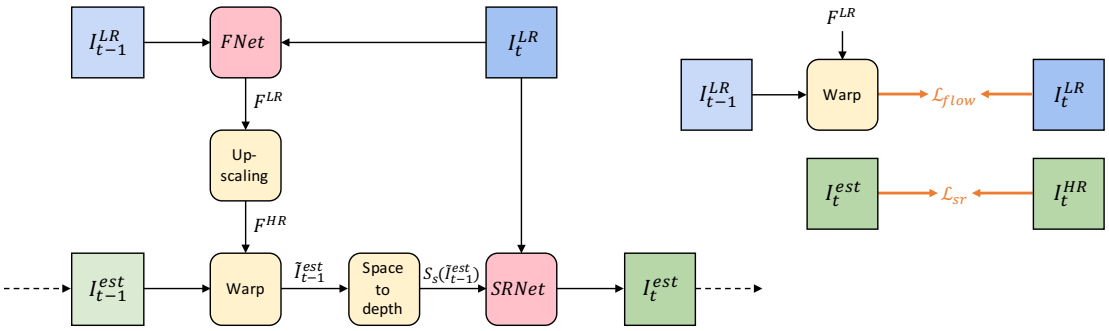


Figure 3.2: Overview of the proposed FRVSR framework (left) and the loss functions used for training (right). After computing the flow F^{LR} in LR space using FNet, we upsample it to F^{HR}. We then use F^{HR} to warp the HR-estimate of the previous frame I_{t-1}^{est} onto the current frame. Finally, we map the warped previous output $\tilde{I}_{t-1}^{\mathrm{est}}$ to LR-space using the space-to-depth transformation and feed it to the super-resolution network SRNet along with the current input frame I_t^{LR}. For training the networks (shown in red), we apply a loss on I_t^{est} as well as an additional loss on the warped previous LR frame to aid FNet.

3.3 Method

After presenting an overview of the FRVSR framework in Section 3.3.1 and defining the loss functions used for training in Section 3.3.2, we justify our design choices in Section 3.3.3 and give details on the implementation and training procedure in Sections 3.3.4 and 3.3.5, respectively.

3.3.1 FRVSR Framework

The proposed framework is illustrated in Figure 3.2. Trainable components (shown in red) include the optical flow estimation network FNet and the super-resolution network SRNet. To produce the HR estimate I_t^{est}, our model makes use of the current LR input frame I_t^{LR}, the previous LR input frame I_{t-1}^{LR}, and the previous HR estimate I_{t-1}^{est}.

1. Flow estimation

As a first step, FNet estimates the flow between the low-resolution inputs I_{t-1}^{LR} and I_t^{LR} yielding the normalized low-resolution flow map

$$F^{\mathrm{LR}} = \mathrm{FNet}(I_{t-1}^{\mathrm{LR}}, I_t^{\mathrm{LR}}) \in [-1,1]^{H \times W \times 2} \tag{3.1}$$

that assigns a position in I_{t-1}^{LR} to each pixel location in I_t^{LR}.

2. Upscaling flow

Treating the flow map F^{LR} as an image, we upscale it using bilinear interpolation with scaling factor s which results in an HR flow-map

$$F^{\mathrm{HR}} = \mathrm{UP}(F^{\mathrm{LR}}) \in [-1,1]^{sH \times sW \times 2}. \tag{3.2}$$

3. Warping previous output

We use the high-resolution flow map F^{HR} to warp the previously estimated image I^{est}_{t-1} according to the optical flow from the previous frame onto the current frame.

$$\tilde{I}^{\mathrm{est}}_{t-1} = \mathrm{WP}(I^{\mathrm{est}}_{t-1}, F^{\mathrm{HR}}) \tag{3.3}$$

We implemented warping as a differentiable function using bilinear interpolation similar to Jaderberg *et al.* (2015).

4. Mapping to LR space

We map the warped previous output $\tilde{I}^{\mathrm{est}}_{t-1}$ to LR space using the space-to-depth transformation

$$\mathrm{S}_s : [0,1]^{sH \times sW \times 3} \to [0,1]^{H \times W \times s^2 C} \tag{3.4}$$

which extracts shifted low-resolution grids from the image and places them into the channel dimension, see Figure 3.3 for an illustration. The operator can be formally described as

$$\mathrm{S}_s(I)_{i,j,k} = I_{si+k\%s,\, sj+(k/s)\%s,\, k/s^2} \tag{3.5}$$

with zero-based indexing, modulus $\%$ and integer division $/$.

5. Super-Resolution

In the final step, we concatenate the LR mapping of the warped previous output $\tilde{I}^{\mathrm{est}}_{t-1}$ with the current low-resolution input frame I^{LR}_t in the channel dimension, and feed the result $I^{\mathrm{LR}}_t \oplus \mathrm{S}_s(\tilde{I}^{\mathrm{est}}_{t-1})$ to the super-resolution network SRNet.

Summary

The final estimate I^{est}_t of the framework is the output of the super-resolution network SRNet:

$$I^{\mathrm{est}}_t = \mathrm{SRNet}(I^{\mathrm{LR}}_t \oplus \mathrm{S}_s(\mathrm{WP}(I^{\mathrm{est}}_{t-1}, \mathrm{UP}(\mathrm{FNet}(I^{\mathrm{LR}}_{t-1}, I^{\mathrm{LR}}_t))))) \tag{3.6}$$

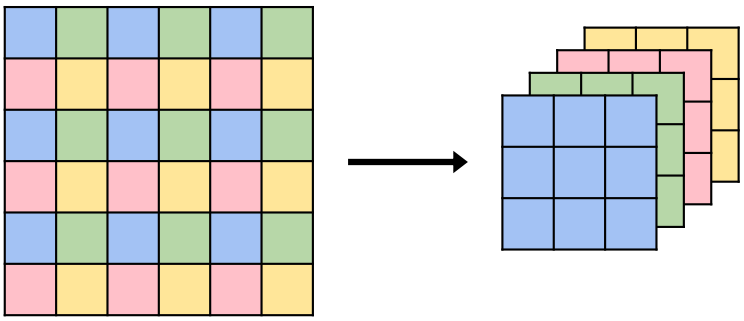


Figure 3.3: Illustration of the space-to-depth transformation S_2. Regular LR grids with varying offsets are extracted from an HR image and placed into the channel dimension, see Equation 3.5 for a formal definition.

3.3.2 Loss Functions

We use two loss terms to train our model, see Figure 3.2, right. The loss $\mathcal{L}_{\text{sr}}$ is applied on the output of SRNet and is backpropagated through both SRNet and FNet:

$$\mathcal{L}_{\text{sr}} = ||I_t^{\text{est}} - I_t^{\text{HR}}||_2^2 \tag{3.7}$$

Since we do not have a ground truth optical flow for our video dataset, we calculate the spatial mean squared error on the warped LR input frames leading to the auxiliary loss term $\mathcal{L}_{\text{flow}}$ to aid FNet during training.

$$\mathcal{L}_{\text{flow}} = ||\,\text{WP}(I_{t-1}^{\text{LR}}, F^{\text{LR}}) - I_t^{\text{LR}}||_2^2 \tag{3.8}$$

The total loss used for training is $\mathcal{L} = \mathcal{L}_{\text{sr}} + \mathcal{L}_{\text{flow}}$.

3.3.3 Justifications

The proposed FRVSR framework is motivated by the following ideas:

• Processing the input video frames more than once leads to high computational cost. Hence, we avoid the sliding window approach and process each input frame only once.

• Having direct access to the previous output can help the network to produce a temporally consistent estimate for the following frame. Furthermore, through a recurrent architecture, the network can effectively use a large number of previous LR frames to estimate the HR frame (see Section 3.4.6) without tradeoffs in computational efficiency. For this reason, we warp the previous HR estimate and feed it to the super-resolution network.

• All computationally intensive operations should be performed in LR space. To this end, we map the previous HR estimate to LR space using the space-to-depth transformation, the inverse of which has been previously used by Shi *et al.* (2016a) for upsampling. Running SRNet in LR space has the additional advantages of reducing the memory footprint and increasing the receptive field when compared to a super-resolution network that would operate in HR space.

3.3.4 Implementation

The proposed model in Figure 3.2 is a flexible framework that leaves the choice for a specific network architecture open. For our experiments, we use fully convolutional architectures for both FNet and SRNet, see Figure 3.4 for details. The design of our optical flow network FNet follows a simple encoder-decoder style architecture to increase the receptive field of the convolutions. For SRNet, we follow the residual architecture used by Sajjadi *et al.* (2017), but replace the upsampling layers with transposed convolutions. Our choice of network architectures strikes a balance between quality and complexity. More recent methods for each subtask, especially more complex optical flow estimation methods (Dosovitskiy *et al.*, 2015; Ilg *et al.*, 2017; Ranjan and Black, 2017) can be easily incorporated and will lead to even better results.

3.3.5 Training and Inference

Our training dataset consists of 40 high-resolution videos (720p, 1080p and 4k) downloaded from vimeo.com. We downsample the original videos by a factor of 2 to have a clean high-resolution ground truth and extract patches of size 256×256 to generate the HR videos. To produce the input LR videos, we apply Gaussian blur to the HR frames and downscale them by sampling every 4-th pixel in each dimension for $s = 4$. Unless specified otherwise, we use a Gaussian blur with standard deviation $\sigma = 1.5$ (see Section 3.4.2).

To train the recurrent system, we extract clips of 10 consecutive frames from the videos using FFmpeg. We avoid cuts or large scene changes in the clips by making sure that the clips do not contain keyframes. All losses are backpropagated through both networks SRNet and FNet as well as through time, *i.e.*, even the optical flow network for the first frame in a clip receives gradients from the super-resolution loss on the 10th frame. The model directly estimates the full RGB video frames, so no post-processing is necessary.

To estimate the first frame I_1^{est} in each clip, we initialize the previous estimate with a black image $I_0^{\text{est}} = 0$ at both training and testing time. The network will then simply upsample the input frame I_1^{LR} independently without additional prior data, similar to a single image super-resolution network. This has the additional benefit of encouraging the network to learn how to upsample single images independently early on during training instead of only relying on copying the previously generated image $\tilde{I}_{t-1}^{\text{est}}$.

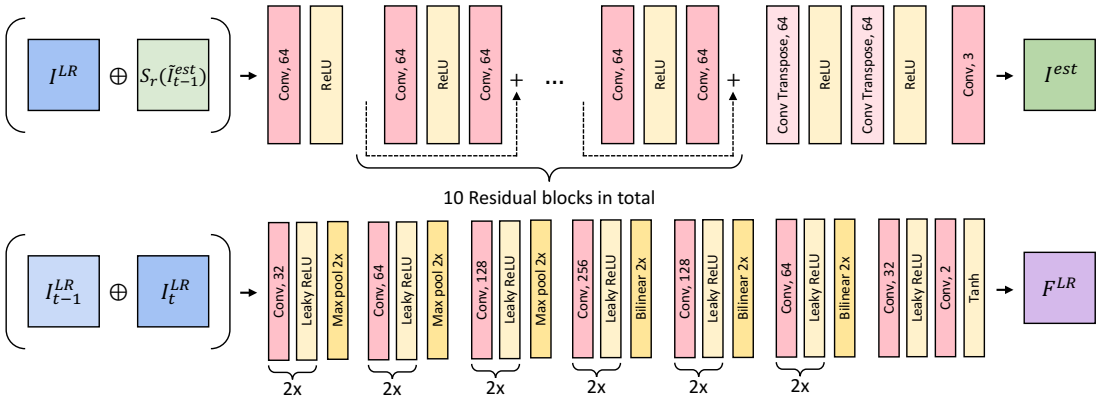


Figure 3.4: Network architectures for SRNet (top) and FNet (bottom) for 4x upsampling. Both networks are fully convolutional and work in LR space. For the inputs, $\oplus$ denotes the concatenation of images in the channel dimension. All convolutions in both networks use 3×3 kernels with stride 1, except for the transposed convolutions in SRNet which use stride 2 for spatial upsampling. The leaky ReLU units in FNet use a leakage factor of 0.2 and the notation 2x indicates that the corresponding block is duplicated.

Our architecture is fully end-to-end trainable and does not require component-wise pre-training. Initializing the networks with the Xavier method (Glorot and Bengio, 2010), we train the model on 2 million batches of size 4 using the Adam optimizer (Kingma and Ba, 2015) with a fixed learning rate of 10^{-4}. Note that each sample in the batch is a set of 10 consecutive video frames, *i.e.*, 40 video frames are passed through the networks in each iteration.

As training progresses, the optical flow estimation gradually improves which gives the super-resolution network higher-quality data to work with, helping it to rely more and more on the warped previous estimate $\tilde{I}^{\text{est}}_{t-1}$. At the same time, the super-resolution network automatically learns to ignore the previous image $\tilde{I}^{\text{est}}_{t-1}$ when the optical flow network cannot find a good correspondence between I^{LR}_{t-1} and I^{LR}_t, *e.g.*, for the very first video frame in each batch or for occluded areas. These cases can be detected by the network through a comparison of the low frequencies in $\tilde{I}^{\text{est}}_{t-1}$ with those in I^{LR}_t. In areas where they do not match, the network ignores the details in $\tilde{I}^{\text{est}}_{t-1}$ and simply upscales the current input frame independently. Once the model has been trained, it can be run on videos of arbitrary size and length due to the fully convolutional nature of the networks. To super-resolve a video, the network is applied frame by frame in a single feed-forward pass. Benchmarks for runtimes of different model sizes are reported in Section 3.4.7.

3.4 Evaluation

For a fair evaluation of the proposed framework on equal ground, we compare our model with two baselines that use the same optical flow and super-resolution networks. After presenting the baselines in Section 3.4.1, we extensively investigate the performance of FRVSR along with the baselines in Section 3.4.2–3.4.7. All experiments are done for the challenging case of 4x upsampling. For evaluation, we use a dataset of ten 3–5s high-quality 1080p video clips downloaded from youtube.com, which we refer to as YT10. Finally, we compare our models with current state-of-the-art methods on the standard Vid4 benchmark dataset (Liu and Sun, 2011) in Section 3.4.8. Following Caballero *et al.* (2017), we compute *video* PSNR on the brightness channel (ITU-R BT.601 YCbCr standard) using the mean squared error over all pixels in the video. For more results and video samples, we refer the reader to the project website at github.com/msmsajjadi/FRVSR.

3.4.1 Baselines

SISR: For the single image super-resolution baseline, we omit optical flow estimation from FRVSR and disregard any prior information, feeding only I_t^{LR} into SRNet.

VSR: To compare with the sliding window approach for video super-resolution, we include this baseline in which a fixed number of input frames are processed to produce a single output frame. Following Kappeler *et al.* (2016) and Caballero *et al.* (2017), we warp the previous and next input frames onto the current frame, concatenate all three frames and feed them to SRNet. Note that this model is computationally more expensive than FRVSR since it runs FNet twice for each frame while the computation for SRNet is almost identical to that of FRVSR.

As with FRVSR, both baselines are trained starting from a Xavier initialization (Glorot and Bengio, 2010) using the Adam optimizer (Kingma and Ba, 2015) with a fixed learning rate of 10^{-4}. We trained the SISR network for 500K steps and VSR for 2 million steps, both using a batch size of 16. All networks are trained using the same dataset, and their losses on a validation dataset have converged at the end of the training.

3.4.2 Blur Size

As mentioned in Section 3.3.5, we apply Gaussian blur to the HR frames before downsampling them to generate the LR input for the network. While a smaller blur kernel results in aliasing, excessive blur leads to loss of high-frequency information in the input, making it harder to reconstruct finer details. To analyze how different approaches perform for blurry or aliased inputs, we trained SISR, VSR and FRVSR on video frames that have been downscaled using different values of standard deviation for the Gaussian blur ranging from $\sigma=0$ to $\sigma=5$, see Figure 3.5. The proposed framework FRVSR significantly outperforms SISR and VSR on all blur sizes. It is interesting to note that SISR, which

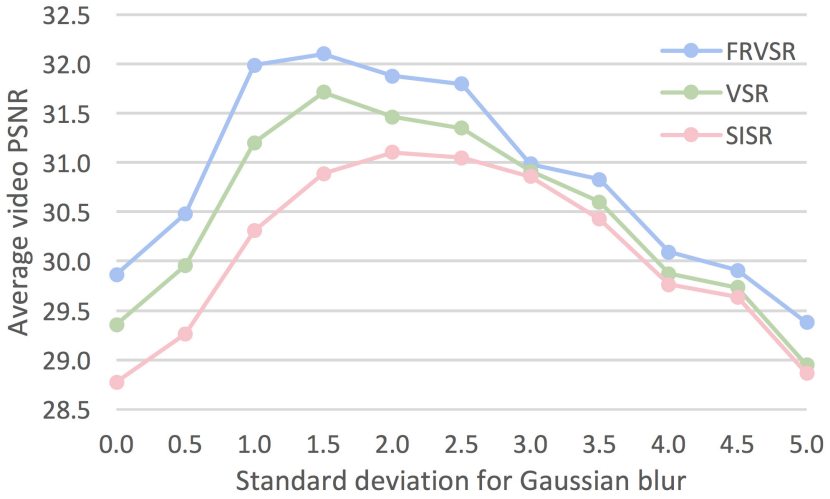


Figure 3.5: Performance for different blur sizes on YT10. For all blur sizes, FRVSR gives the best results. The best PSNR of FRVSR ($\sigma = 1.5$) is 1.00 dB and 0.39 dB higher than the best of SISR ($\sigma = 2.0$) and VSR ($\sigma = 1.5$), respectively.

relies on a single LR image for upsampling, benefits the most from larger blur kernels compared to VSR and FRVSR which perform best with $\sigma = 1.5$. This is due to the fact that video super-resolution methods are able to blend information from multiple frames and therefore benefit from sharper inputs. In the remaining experiments, we use $\sigma = 1.5$.

3.4.3 Training Clip Length

Since FRVSR is a recurrent network, it can be trained on video clips of any length. To test the effect of the clip length used to train the network, we trained the same model using video clips of length 2, 5 and 10, yielding average video PSNR values of 31.60, 32.01 and 32.10 on YT10, respectively. This shows that the PSNR has already started to saturate with a clip length of 5 and going beyond 10 may not yield significant improvements.

3.4.4 Degraded Inputs

To see how different models perform under input degradations, we trained and evaluated FRVSR and the baselines using noisy and compressed input frames. Table 3.1 shows the performance of these models on YT10 for varying levels of Gaussian noise and JPEG compression quality. The proposed framework consistently outperforms both SISR and VSR by 0.36–0.91 dB and 0.18–0.48 dB, respectively.

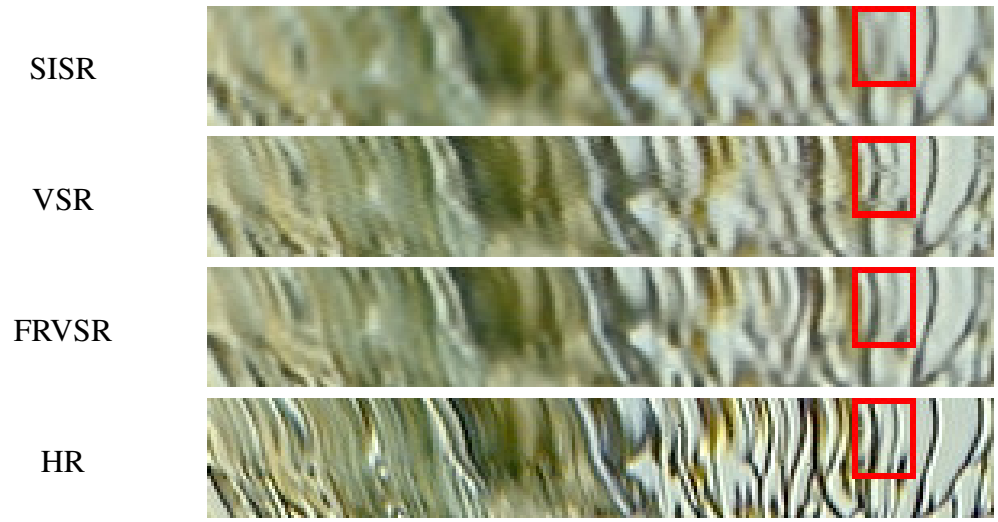


Figure 3.6: Temporal profiles for *Calendar* from Vid4. VSR yields finer details than SISR, but it's output still contains temporal inconsistencies (*e.g.*, in the red boxes). Only FRVSR is able to produce temporally consistent results while reproducing fine details.

model	$\sigma = 0.025$	$\sigma = 0.075$	JPG 40	JPG 70
SISR	29.93	28.20	27.94	28.88
VSR	30.36	28.42	28.12	29.07
FRVSR	**30.84**	**28.62**	**28.30**	**29.29**

Table 3.1: Average video PSNR of various models under Gaussian noise (left) and JPEG artifacts (right) on YT10. In all experiments, FRVSR achieves the highest PSNR.

3.4.5 Temporal Consistency

Analyzing the temporal consistency of the results is best done by visual inspection of the video results. However, to compare the results on paper, we follow Caballero *et al.* (2017) and show *temporal profiles*, see Figure 3.6. A temporal profile is generated by taking the same horizontal row of pixels from a number of frames in the video and stacking them vertically into a new image. Flickering in the video will show up as jitter and jagged lines in the temporal profile. SISR produces blurry images that are temporally inconsistent. While VSR produces sharper results than SISR, it still has significant flickering artifacts since each output frame is estimated separately. In contrast, FRVSR produces the most consistent results while containing even finer details in each image.

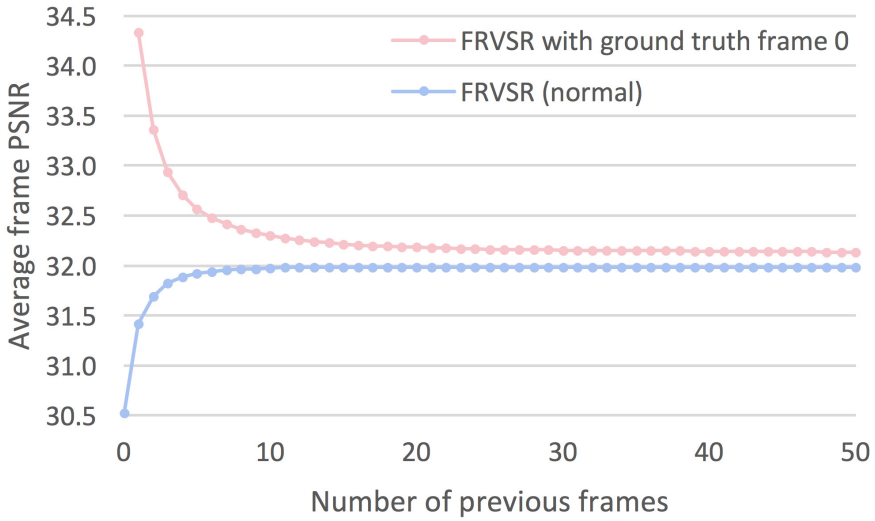


Figure 3.7: Performance of FRVSR on YT10 as a function of the number of previous frames processed. In the normal mode (blue), PSNR increases up to 12 frames, after which it remains stable. When the first HR image is given (red), FRVSR propagates high-frequency details across a large number of frames and performs better than the normal mode even after 50 frames.

3.4.6 Range of Information Flow

Existing approaches to video super-resolution often use a fixed number of input frames to produce a single output frame (usually 3 to 7 frames). Increasing this number increases the maximum number of frames over which details can be propagated. While this can result in higher-quality videos, it also substantially increases the computational cost, leading to a tradeoff between efficiency and quality. In contrast, due to its recurrent nature, FRVSR can pass information across a large number of frames without increasing computations. Figure 3.7 shows the performance of FRVSR as a function of the number of frames processed. In the normal mode (blue curve) in which a black frame is used as the first frame's previous HR estimate, the performance steadily improves as more frames are processed and it plateaus at 12 frames. When we replace the first previous HR estimate with the corresponding groundtruth HR frame (red curve), FRVSR carries the high-frequency details across a large number of frames and performs better than the normal mode even after 50 frames.

To investigate the maximum effective range of information flow, we start the same model at different input frames in the same video and compare the performance. Figure 3.8 shows such a comparison for the *Foliage* video from Vid4. As we can see, the gap between the curves for the models that start at frame 1 and frame 11 only closes towards the end of the clip, showing that FRVSR is propagating information over more than 30 frames.

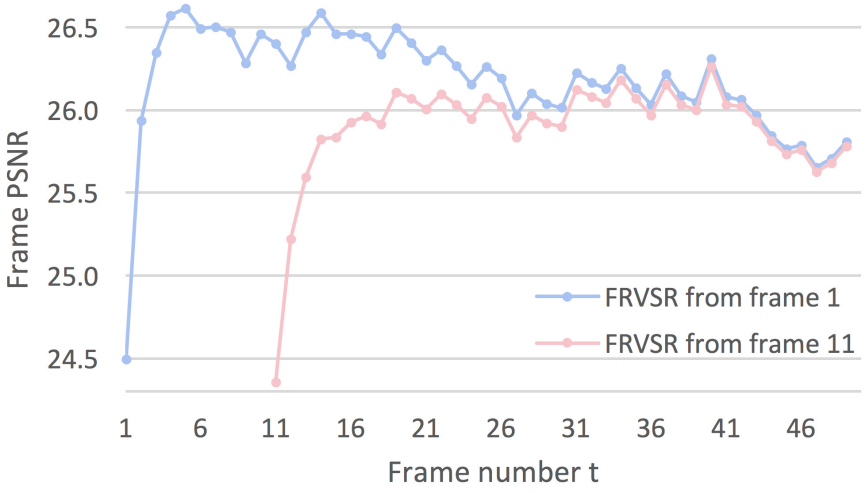


Figure 3.8: Performance of FRVSR started at the 1st and 11th frame of *Foliage* from Vid4. The gap between the curves only closes towards the end of the clip, showing FRVSR's ability to retain details over a large range of video frames.

To propagate details over such a large range, previous state-of-the-art methods (Kappeler *et al.*, 2016; Caballero *et al.*, 2017; Tao *et al.*, 2017; Liu *et al.*, 2017; Makansi *et al.*, 2017) would have to process an inhibiting number of input frames for each output image, which would be computationally infeasible.

3.4.7 Network Size and Computational Efficiency

To see how the performance of different models varies with the size of the network, we trained and evaluated FRVSR and the baselines with different numbers of residual blocks and convolution filters in SRNet, see Figure 3.9. It is interesting to note that the video super-resolution models FRVSR and VSR clearly benefit from larger models while the performance of SISR does not change significantly beyond 5 residual blocks. We can also see that FRVSR achieves better results than VSR despite being faster: The FRVSR models with 5 residual blocks outperform the VSR models with 10 residual blocks, and the FRVSR models with 3 residual blocks outperform the VSR models with 5 residual blocks for the same number of convolution filters. In our TensorFlow implementation on an Nvidia P100, producing a single full HD frame for 4x upscaling takes 74ms for FRVSR with 3 residual blocks and 64 filters, and 191ms with 10 blocks and 128 filters.

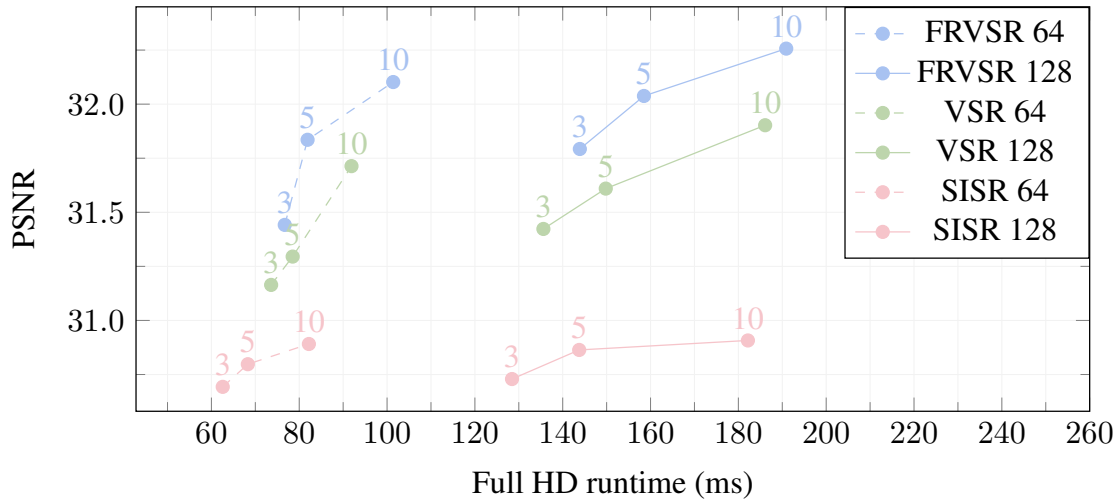


Figure 3.9: Performance on YT10 for different numbers of convolution filters (64 / 128) and residual blocks in SRNet. FRVSR achieves better results than both baselines with significantly smaller super-resolution networks and less computation time. For example, FRVSR with 5 residual blocks is both faster and better than VSR with 10 residual blocks.

3.4.8 Comparison with Prior Art

Table 3.2 compares the proposed FRVSR approach with various state-of-the-art video super-resolution approaches on the standard Vid4 benchmark dataset by PSNR and SSIM. We report results for two FRVSR networks: FRVSR 10-128, which is our best model with 10 residual blocks and 128 convolution filters, and FRVSR 3-64, which is our most efficient model with only 3 residual blocks and 64 convolution filters. For the baselines SISR and VSR, we report their best results which correspond to 10 residual blocks and 128 convolution filters. We also include RAISR (Romano *et al.*, 2016) as an off-the-shelf single image super-resolution alternative. For all competing methods except (Huang *et al.*, 2015b; Romano *et al.*, 2016; Caballero *et al.*, 2017), we used the output images provided by the corresponding authors to compute PSNR and SSIM. We did not use the first and last two frames in our evaluation since Liu *et al.* (2017) do not produce outputs for these frames. Also, for each video, we removed border regions such that the LR input image is a multiple of 8. For (Huang *et al.*, 2015b; Caballero *et al.*, 2017), we use the PSNR and SSIM values reported in the respective publications since we could not confirm them independently. For (Romano *et al.*, 2016), we used the models provided by the authors to generate the output images.

As shown in Table 3.2, FRVSR outperforms the current state of the art by more than 0.5 dB. In fact, even our most efficient model FRVSR 3-64 produces state-of-the-art results by PSNR and beats all previous neural network-based methods by SSIM. It it interesting that our small model, despite being much more efficient, produces results that are very

Figure 3.10: Visual comparison with previous methods on *Foliage* from Vid4. Amongst prior art, Liu and Sun recover the finest details, but their result has blocky artifacts, and their method uses a slow optimization procedure. Between the remaining methods, even the result of our smallest model FRVSR 3-64 is sharper and contains more details than prior art, producing results similar to the much bigger VSR model. Our larger model FRVSR 10-128 recovers the most accurate image.

close to the much larger model VSR 10-128 on the Vid4 dataset. Figure 3.10 shows a visual comparison of the different approaches. We can see that our models are able to recover fine details and produce visually pleasing results. Even our most efficient network FRVSR 3-64 produces higher-quality results than prior art.

3.5 Future Work

Since our framework relies on the HR estimate I^{est} for propagating information, it can reconstruct details and propagate them over a large number of frames (see Section 3.4.6). At the same time, any detail can only persist in the system as long as it is contained in I^{est}, as it is the only way through which SRNet can pass information to future iterations. Due to the spatial loss on I^{est}, SRNet has no way to pass on auxiliary information that could potentially be useful for future frames in the video, *e.g.*, for occluded regions. As a result, occlusions irreversibly destroy all previously aggregated details in the affected areas and the best our model can do for the previously occluded areas is to match the performance of single image super-resolution models. In contrast, models that use a fixed number of input frames can still combine information from frames that do not have occlusions to produce better results in these areas. To address this limitation, it is natural to extend the framework with an additional memory channel. However, preliminary experiments in this direction with both static and motion-compensated memory did not improve the performance of the architecture, so we leave these extensions to future work.

Method	Bicubic	RAISR	BRCN	VESPCN	$B_{1,2,3}+T$	DRVSR	Bayesian
PSNR	23.53	24.24	24.43*	25.35*	25.35	25.87	26.16
SSIM	0.628	0.665	0.662*	0.756*	0.738	0.772	0.815

Method	SISR (10-128)	VSR (10-128)	FRVSR (3-64)	FRVSR (10-128)
PSNR	24.96	26.25	26.17	**26.69**
SSIM	0.721	0.803	0.798	**0.822**

Table 3.2: Comparison of average PSNR and SSIM on the standard Vid4 dataset for scaling factor $s=4$. We compare our models with RAISR (Romano *et al.*, 2016), BRCN (Huang *et al.*, 2015b), VESPCN (Caballero *et al.*, 2017), $B_{1,2,3}+T$ (Liu *et al.*, 2017), DRVSR (Tao *et al.*, 2017), and Bayesian (Liu and Sun, 2011). Our smallest model FRVSR 3-64 already produces better results than all prior art including the computationally expensive optimization-based method by Liu and Sun (2011) by PSNR. Using a bigger super-resolution network helps FRVSR 10-128 to add an additional 0.5 dB on top and achieve state-of-the-art results by SSIM as well, showing that the proposed framework can greatly benefit from more powerful networks. Values marked with a star have been copied from the respective publications.

Since the model is conceptually flexible, it can be easily extended to other applications. As an example, one may plug in the original HR frame I^{HR}_{t-1} in place of the estimated frame I^{est}_{t-1} for every K-th frame. This could enable an efficient video compression method where only one in K HR-frames needs to be stored while the remaining frames would be reconstructed by the model.

A further extension of our framework would be the inclusion of more advanced loss terms which have recently been shown to produce more visually pleasing results (Ledig *et al.*, 2017; Sajjadi *et al.*, 2017). The recurrent architecture in FRVSR naturally encourages the network to produce temporally consistent results, making it an ideal candidate for further research in this direction.

3.6 Summary

We propose a flexible end-to-end trainable framework for video super-resolution that is able to generate higher quality results while being more efficient than existing sliding window approaches. In an extensive set of experiments, we show that our model outperforms competing baselines in various different settings. The proposed model also significantly outperforms state-of-the-art video super-resolution approaches both quantitatively and qualitatively on a standard benchmark dataset.

Chapter 4

Tempered Adversarial Networks

Generative adversarial networks (GANs) have been shown to produce realistic samples from high-dimensional distributions, but training them is considered hard. A possible explanation for training instabilities is the inherent imbalance between the networks: While the discriminator is trained directly on both real and fake samples, the generator only has control over the fake samples it produces since the real data distribution is fixed by the choice of a given dataset. We propose a simple modification that gives the generator control over the real samples which leads to a tempered learning process for both generator and discriminator. The real data distribution passes through a *lens* before being revealed to the discriminator, balancing the generator and discriminator by gradually revealing more detailed features necessary to produce high-quality results. The proposed module automatically adjusts the learning process to the current strength of the networks, yet is generic and easy to add to any GAN variant. In a number of experiments, we show that this can improve quality, stability and/or convergence speed across a range of different GAN architectures (DCGAN, LSGAN, WGAN-GP).

4.1 Introduction

Generative Adversarial Networks (GANs) have been introduced as the state of the art in generative models (Goodfellow *et al.*, 2014). They have been shown to produce sharp and realistic images with fine details (Denton *et al.*, 2015; Radford *et al.*, 2016b; Chen *et al.*, 2016; Zhang *et al.*, 2017b). The basic setup of GANs is to train a parametric nonlinear function, the *generator G*, which maps samples from random noise drawn from a distribution $\mathcal{Z}$ into samples of a fake distribution $G(\mathcal{Z})$ which are close in terms of some measure to a real world empirical data distribution $\mathcal{X}$. To achieve this goal, a *discriminator D* is trained to provide feedback in the form of gradients for the generator. This feedback can be the confidence of a classifier discriminating between real and fake examples (Goodfellow *et al.*, 2014; Mao *et al.*, 2017; Arjovsky *et al.*, 2017; Gulrajani *et al.*, 2017) or an energy defined in terms of a reconstruction loss of an autoencoder (Zhao *et al.*, 2017a; Berthelot *et al.*, 2017).

GANs are infamous for being difficult to train and sensitive to small changes in hyperparameters (Goodfellow *et al.*, 2016). A typical source of instability is the discriminator

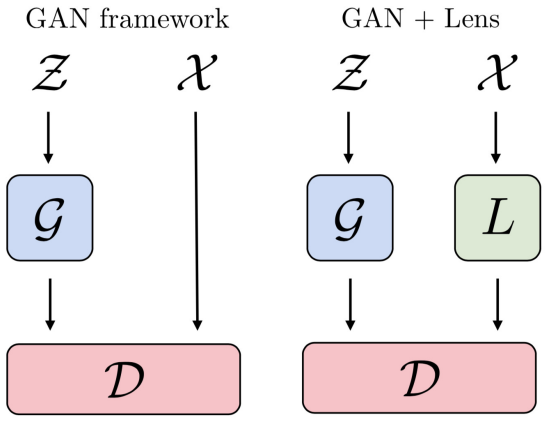


Figure 4.1: Schematic of the proposed module. We add a lens L in between the real data $\mathcal{X}$ and the discriminator D. The lens is compatible with any type of GAN and dataset type. It finds a balance between fooling the discriminator and a reconstruction loss, leading to a tempered training procedure that self-adjusts to the capabilities of the current generator w.r.t. the current discriminator.

rapidly overpowering the generator which leads to problems such as vanishing gradients or mode collapse. In this case, $G(\mathcal{X})$ and $\mathcal{X}$ are too distant from each other and the discriminator learns to fully distinguish them (Arjovsky and Bottou, 2017). While several GAN variants have been introduced to address the problems encountered during training (Berthelot *et al.*, 2017; Zhao *et al.*, 2017a; Arjovsky *et al.*, 2017; Gulrajani *et al.*, 2017), finding stable and more reliable training procedures for GANs is still an open research question (Lucic *et al.*, 2018).

Our Contributions

In this work we propose a general and dynamic, yet simple to implement extension to GANs that encourages a smoother training procedure. We introduce a *lens* module L which gives the generator control over the real data distribution $\mathcal{X}$ before it enters the discriminator. By adding the lens between the real data samples and the discriminator, we allow training to self-stabilize by automatically balancing a reconstruction loss with the current performance of the generator and discriminator. For instance, a lens could implement an image blurring operation which gradually gets reduced during training, thus only requiring the generation of good blurry images at the beginning, which gradually become sharper during training. While this analogy from optics motivates the term lens, in practice we learn the lens from data as explained below.

While the generator in a regular GAN chases a fixed distribution $\mathcal{X}$, the proposed lens moves the target distribution closer to the generated samples $G(\mathcal{Z})$ which leads to a better

optimization behavior.

4.2 Tempered Adversarial Networks

The original formulation for GANs poses the training process as a minimax game between the generator G and discriminator D over the value function $\mathcal{V}$:

$$\min_G \max_D \mathcal{V}(D,G) = \mathbb{E}_{x\sim\mathcal{X}}[\log(D(x))] + \mathbb{E}_{z\sim\mathcal{Z}}[1-\log(D(G(z)))] \tag{4.1}$$

In practice, both generator and discriminator are implemented as neural networks. The generator maps a random distribution $\mathcal{Z}$ to $G(\mathcal{Z})$ which is in the same space as the real data distribution $\mathcal{X}$. While the discriminator sees both real samples from $\mathcal{X}$ and fake samples from $G(\mathcal{Z})$, the generator only has control over the samples it produces itself, *i.e.*, it has no control over the real data distribution $\mathcal{X}$ which is fixed throughout training. To resolve this asymmetry, we add a lens module L which modifies the real data distribution $\mathcal{X}$ before it is passed to the discriminator.

In practice, we use a neural network for L. The only change in the GAN architecture is consequently the input to the discriminator, which changes from $\{\mathcal{X}, G(\mathcal{X})\}$ to $\{L(\mathcal{X}), G(\mathcal{X})\}$.

We train the lens with two loss terms: an adversarial loss $\mathcal{L}_L^A$ and a reconstruction loss $\mathcal{L}_L^R$. The adversarial loss is supposed to maximize the loss of the discriminator of the respective GAN architecture, *i.e.*, $\mathcal{L}_L^A \approx -\mathcal{L}_D$. For the specific loss functions we used with the different GAN variants, see Sections 4.2.1–4.2.3.

Additionally, we add a reconstruction loss to prevent the lens from converging to trivial solutions (*e.g.*, mapping all samples to zero):

$$\mathcal{L}_L^R = ||\mathcal{X} - L(\mathcal{X})||_2^2 \tag{4.2}$$

The overall loss for the lens is

$$\mathcal{L}_L = \lambda \mathcal{L}_L^A + \mathcal{L}_L^R \tag{4.3}$$

The lens can automatically balance a good reconstruction of the original samples with the objective of mapping the real data distribution $\mathcal{X}$ close to the generated data distribution $G(\mathcal{Z})$ w.r.t. the probabilities given by the discriminator. As training progresses, the generated samples get closer to the real samples, *i.e.*, the lens can afford to reconstruct the real data samples better. Once the discriminator starts to see differences, the loss term $\mathcal{L}_L^A$ increases which makes L shift the real data distribution $\mathcal{X}$ towards the generated samples, helping to keep $G(\mathcal{Z})$ and $L(\mathcal{X})$ closer together which yields better gradients during training.

To accelerate this procedure, we set $\lambda = 1$ at the beginning of the training procedure and then gradually decrease it to $\lambda = 0$, at which point L is only trained with $\mathcal{L}_L^R$, forcing

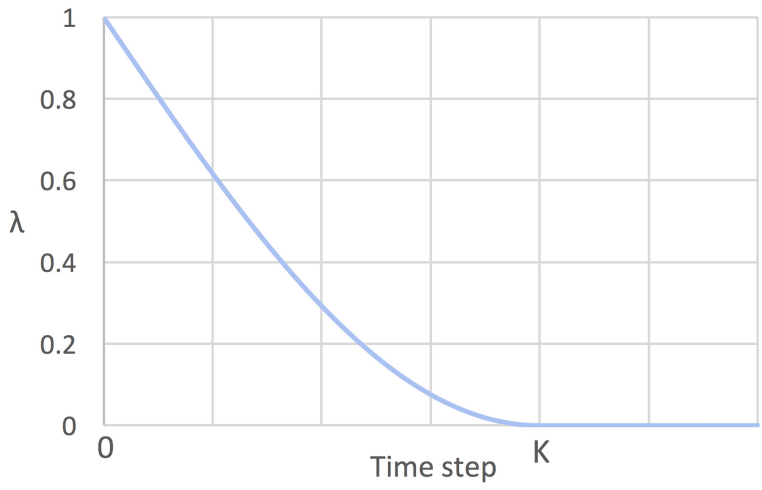


Figure 4.2: Schedule for the weight λ for the adversarial loss term $\mathcal{L}_L^A$ of the lens during training. As training progresses, the value is lowered in a smooth way from 1 to 0 in K steps, increasing the relative weight of the reconstruction loss for the lens. We set K=10k in all experiments. While lower values showed faster convergence rates in our experiments, we opted for a single value in all experiments for simplicity and to avoid adding yet another hyperparameter that needs to be tuned. We found that the performance is robust against changes for the specific value for K and that a single value yields good results across datasets and GAN architectures.

it to converge to the identity mapping $L(\mathcal{X}) = \mathcal{X}$. To have a smooth transition from adversarial samples $L(\mathcal{X})$ to the real data distribution $\mathcal{X}$ in K steps, we adapt the value for λ as

$$\lambda = \begin{cases} 1 - \sin(t\pi/2K), & t \leq K \\ 0, & t > K \end{cases} \tag{4.4}$$

for the t-th time step during training. The value of λ over time can be seen in Figure 4.2. Once the lens converges to the identity mapping, training reduces to the original GAN architecture without a lens. In all experiments, we set $K = 10^5$ unless specified otherwise. Lower values for K lead to faster convergence, but to avoid introducing a new hyperparameter that needs to be tuned, and for simplicity, we choose the same value for all experiments. Note that this choice is clearly not optimal for all tasks and tuning the value can easily lead to even faster convergence and higher quality samples.

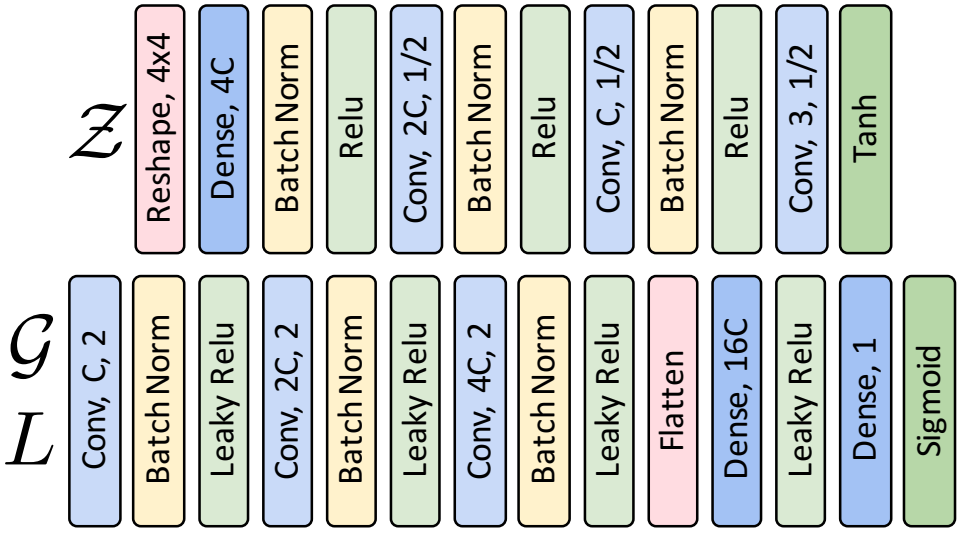


Figure 4.3: Network architecture of the generator G (top) and discriminator D (bottom). The design follows Radford *et al.* (2016b). The strides of the convolutions are 1/2 for upsampling in G and 2 for downsampling in D. The kernel size is 4×4 in both networks. The number of parameters can be varied by adjusting C.

4.2.1 Objectives for Classical GAN Formulation

In the original work, Goodfellow *et al.* (2014) use the loss

$$\mathcal{L}_G = -\log(D(G(\mathcal{Z}))) \tag{4.5}$$

for the generator, and

$$\mathcal{L}_D^{original} = -\log(D(\mathcal{X})) - \log(1 - D(G(\mathcal{Z}))) \tag{4.6}$$

for the discriminator. The objectives of generator and discriminator remain unchanged, though the input of the real data to D is changed from $\mathcal{X}$ to $L(\mathcal{X})$:

$$\mathcal{L}_D = -\log(D(L(\mathcal{X}))) - \log(1 - D(G(\mathcal{Z}))). \tag{4.7}$$

The lens is trained against the discriminator with the adversarial loss term

$$\mathcal{L}_L^A = -\log(1 - D(L(\mathcal{X}))) \tag{4.8}$$

which minimizes the output of the discriminator for the lensed data points using the nonsaturating loss.

4.2.2 Objectives for LSGAN

In LSGAN (Mao *et al.*, 2017), the log-loss is replaced by the squared distance. This leads to the adversarial loss

$$\mathcal{L}_G = ||D(G(\mathcal{Z})) - 1||_2^2 \tag{4.9}$$

for the generator, and

$$\mathcal{L}_D = ||D(G(\mathcal{Z}))||_2^2 + ||D(L(\mathcal{X})) - 1||_2^2 \tag{4.10}$$

for the discriminator. The lens works against the discriminator with the adversarial loss

$$\mathcal{L}_L^A = ||D(L(\mathcal{X}))||_2^2 \tag{4.11}$$

4.2.3 Objectives for WGAN-GP

The discriminator or *critic* in the WGAN-GP variant (Gulrajani *et al.*, 2017) outputs values that are unbounded, *i.e.*, there is no sigmoid activation at the after the last dense layer in Figure 4.3. The objectives are

$$\mathcal{L}_G = -D(G(\mathcal{Z})) \tag{4.12}$$

for the generator, and

$$\mathcal{L}_D = D(G(\mathcal{Z})) - D(L(\mathcal{X})) \tag{4.13}$$

for the critic. Again, the lens works against the critic, so we use the adversarial objective

$$\mathcal{L}_L^A = D(L(\mathcal{X})) \tag{4.14}$$

for the lens for this GAN variant.

4.2.4 Architecture, Training and Evaluation metrics

The lens can be any function which maps from the usually high-dimensional space of the real data distribution $\mathcal{X}$ to itself. Note that the lens does not need to be injective – in fact, early on during training, mapping several different points to the same data point can be a simple way to decrease the complexity of the data distribution which will likely decrease the loss term $\mathcal{L}_L^A$. Since it is desirable for the lens to turn into the identity mapping at some point during training, we have chosen a residual fully convolutional neural network architecture for the lens, see Figure 4.4.

The network architecture and training procedure for the generator and discriminator depend on the chosen GAN framework. For the experiments with the original GAN loss, we use the DCGAN architecture along with its common tweaks (Radford *et al.*, 2016b), namely, strided convolutions instead of pooling layers, applying batch normalization

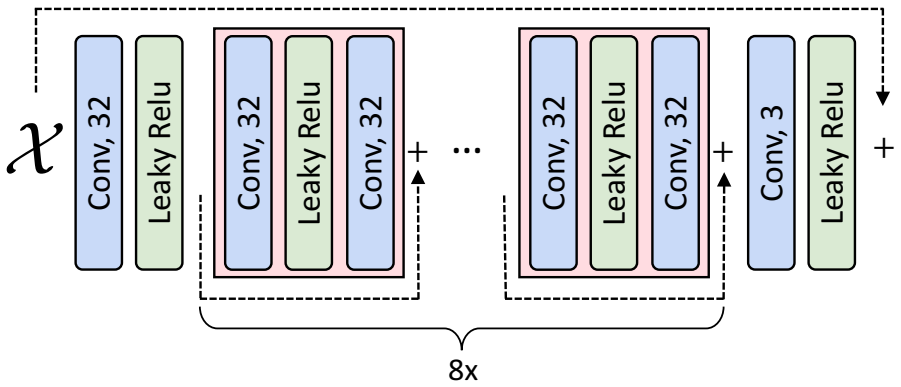


Figure 4.4: Network architecture of the proposed lens that is similar to Sajjadi *et al.* (2017). The core of the network is composed of 8 residual blocks. To help convergence to identity, we add an additional residual connection from the input to the output. All convolutions have 3×3 kernels and stride 1.

in both networks, using ReLU in the generator and leaky ReLU in the discriminator, and Adam (Kingma and Ba, 2015) as the optimizer. See Figure 4.3 for an overview of the networks. LSGAN is trained in the same setting but without batchnorm. For the WGAN-GP experiments, we used the implementation from Gulrajani (2017) which uses very similar models but the RMSProp optimizer (Hinton *et al.*, 2012). We train the lens alongside the generator and discriminator and update it once per iteration regardless of the GAN variant. Note that the networks for the DCGAN and LSGAN experiments have intentionally been chosen not to have a very large number of feature channels to avoid memorization on small datasets which is why the results on an absolute scale are certainly not state of the art. We train using batch sizes of 32 and 64, a learning rate of 10^{-4} and we initialize the networks with the Xavier initialization (Glorot and Bengio, 2010).

For quantitative evaluation, previous works have been reporting the Inception score (Salimans *et al.*, 2016a), though its accuracy has been questioned (Barratt and Sharma, 2018). Recently, the *Fréchet Inception Distance* (FID) has been shown to correlate well with the perceived quality of samples, so we follow Heusel *et al.* (2017) and report FID scores. Note that a lower FID is better. For computational reasons, the FID scores are computed on sets of 4096 samples for the DCGAN and LSGAN experiments. While this is lower than the recommended 10k and should therefore not be compared directly with other publications, we found the sample size to be sufficient to capture relative improvements as long as sample sizes are identical. For the WGAN-GP experiments, we used sample sizes of 10k data points. The image size in all experiments is 32×32 pixels with 1 color channel for MNIST and 3 color channels for all other experiments.

4.3 Related Work

After its introduction (Goodfellow *et al.*, 2014), GANs have received a lot of attention from the community. There are several lines of work to improve the training procedure of GANs. Radford *et al.* (2016b) proposed heuristic guidelines for the design of GAN architectures, *e.g.*, recommending the use of strided convolutions and batch normalization (Ioffe and Szegedy, 2015) in both generator and discriminator. Several works follow this trend, *e.g.*, Salimans *et al.* (2016a) propose the use of further methods to stabilize the performance of GANs including feature matching, historical averaging, minibatch discrimination and one-sided label smoothing (Szegedy *et al.*, 2016). More closely related to our work, Arjovsky and Bottou (2017) propose adding noise to the samples during training with the motivation of increasing the support of the generated and real data distributions which leads to more meaningful gradients. The amount of noise is reduced manually during training. In our work, the lens is not constrained in the mapping that it can apply to balance the training procedure. Furthermore, the effect of the lens is automatically balanced with a reconstruction term that adjusts the intervention of the lens dynamically during training depending on the current balance between generator and discriminator.

There are several works which approach the problem by using multiple networks instead of one. Denton *et al.* (2015) propose a Laplacian pyramid of generator-discriminator pairs for generating images. Zhang *et al.* (2017b) use a similar approach by using one GAN to produce a low-resolution image and another GAN which produces higher-resolution images conditioned on the output of the low-resolution GAN. Such methods have the drawback that several GANs need to be trained which increases the number of parameters and introduces a computational bottleneck. Most recently, Karras *et al.* (2018) produced convincing high-resolution images of faces by first learning the low frequencies in images and then progressively growing both networks to produce higher-resolution images. While promising, all of the methods above are constrained to generating images since the concept of resolution is not easily generalizable to other domains.

Another line of research attacks the problem of training GANs by changing the loss functions, *e.g.*, Mao *et al.* (2017) use the least-squares distance loss whereas Arjovsky *et al.* (2017) approximate the Wasserstein distance which provides more stable gradients for the generator. Gulrajani *et al.* (2017) improve upon the latter by replacing weight clipping in the discriminator with a gradient penalty which accelerates the training procedure considerably.

In the context of training neural networks, Gulcehre *et al.* (2017) smoothen the objective function by adding noise to activation functions and then gradually decrease the level of noise as training progresses. Bengio *et al.* (2009) coin the term *curriculum learning* where the idea is to present the samples during training in a specific order that improves the learning process. Our approach may have a similar effect, but differs in that we present all samples of the original dataset to the networks, modifying them dynamically in a way that stabilizes the learning process.

4.4 Experiments

Showing that modifications or additions to GANs lead to *better* results in any way is a delicate topic that has raised much controversy in the community. Most recently, the findings of Lucic *et al.* (2018) suggest that with a sufficient computational budget, any GAN architecture can be shown to perform at least as well or better than another, if a smaller computational budget is spent on the hyperparameter search for the latter. To avoid this fallacy and to prevent choices such as the network architecture or chosen hyperparameters to favor one or another method, we follow common guidelines that are currently in use for training GANs and we conduct experiments with three different GAN frameworks: the original GAN formulation by Goodfellow *et al.* (2014); LSGAN, where Mao *et al.* (2017) replace the log-loss with the least-squares loss; and WGAN-GP, where Gulrajani *et al.* (2017) minimize the approximated Wasserstein distance between real and generated data distributions and where the training procedure includes a gradient penalty for the discriminator. For the network architecture, we follow standard design patterns (see Section 4.2.4). In our experimental section, we do not strive for state of the art in the end results, but rather we test how much of an effect the lens can have on training. We show that the simple addition of a lens can help improve results across various GAN frameworks. We hope that this insight will help ongoing efforts to understand and improve the training of GANs and other neural network architectures.

In all experiments, the random weights for the initialization of the networks were identical for the GANs with and without a lens. All experiments have further been run with at least 3 different random seeds for the weight initialization to prevent chance from affecting the results.

4.4.1 DCGAN

MNIST

We begin with the original GAN variant on the classical MNIST dataset. To analyze the behavior of the lens, we first consider the case of a fixed $\lambda = 1$, *i.e.*, the lens has no direct incentive to become perfect identity. Figure 4.5 (top) shows generated and lensed samples at different training stages for this architecture. At the beginning of training, the lens scrambles the MNIST digits to look more similar to the generated images. As the generator catches up and produces digit-like samples, the lens can afford to improve reconstruction. Since the lens acts as a balancing factor between the G and D, this leads to a very stable training procedure. However, even after 10M steps, the reconstruction of the lens still improves, as does the FID score of the generated samples (see FID plot in Figure 4.5, bottom left). In comparison, the GAN without a lens converges much faster to better FID scores (Figure 4.5, bottom right, green curve).

To accelerate the training procedure, we adapt the weight of λ as explained in Section 4.2. As this forces the lens to turn into a perfect identity mapping at some point

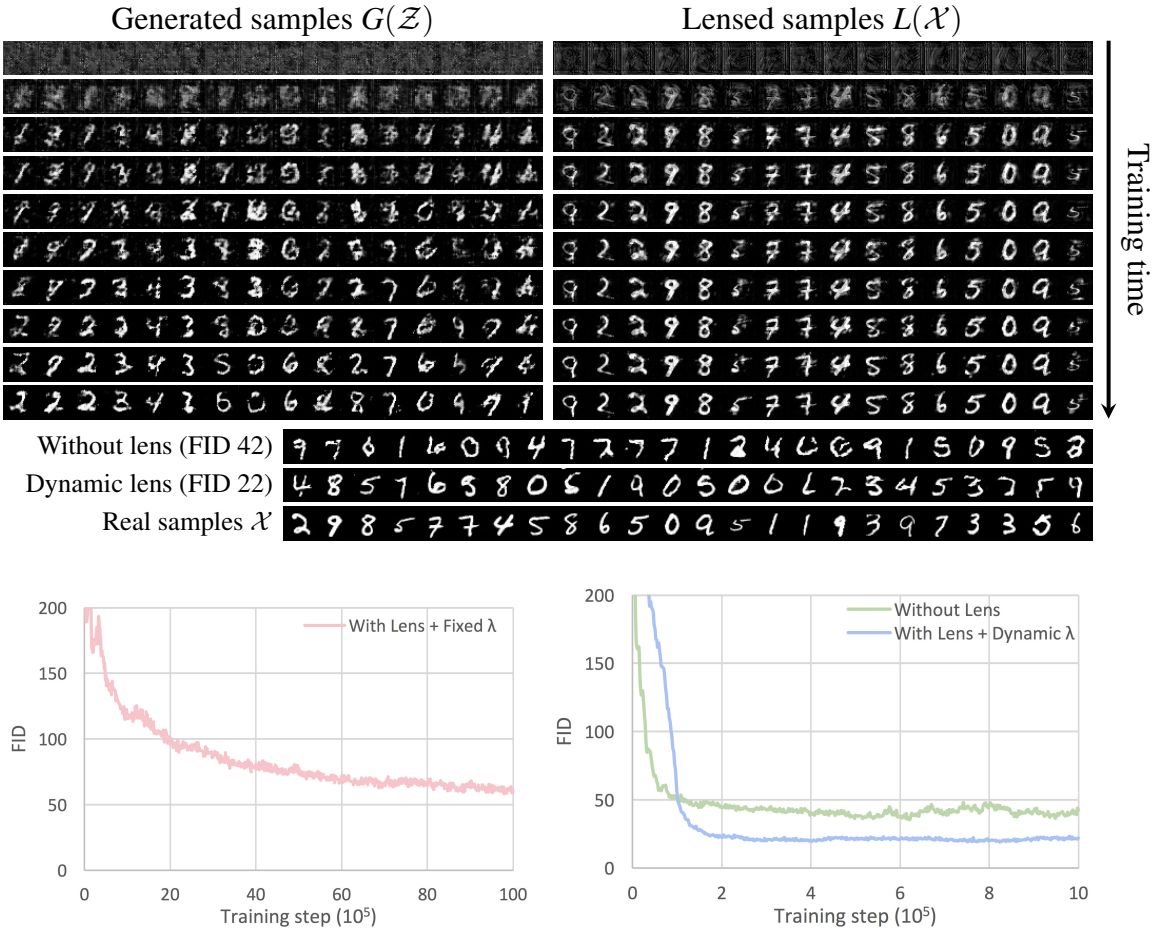


Figure 4.5: MNIST digits produced by DCGAN with a lens with **fixed** $\lambda = 1$ (top). The columns show generated and lensed samples. The lens L adds pertubations that make the real data samples look more similar to fake samples. As training progresses, the quality increases and the reconstruction of L improves steadily. Ideally, the system would converge to a point where G produces samples that are indistinguishable from $\mathcal{X}$ for a fully trained discriminator – at this point, L would turn into the identity mapping. While training with the lens is very stable and while the FID was still decreasing when we stopped training, the reconstructions are not perfect even after 10M training steps and the FID is still only 60, *i.e.*, it has not yet even reached the performance of DCGAN without a lens after only 1M steps (bottom right, green curve). When the value for λ is adapted (see Section 4.2), training is greatly sped up and, the quality of the samples is substantially higher (FID 22) than for the GAN without a lens (FID 42). The difference is also visible in the results, where the GAN with a lens produces better looking MNIST digits. Note that the FID is initially higher for the GAN with a lens in the bottom right. This is because the FID is always measured against the real samples $\mathcal{X}$, while G is initially trained for the lensed distribution $L(\mathcal{X})$ that differs from $\mathcal{X}$ in the early training stages.

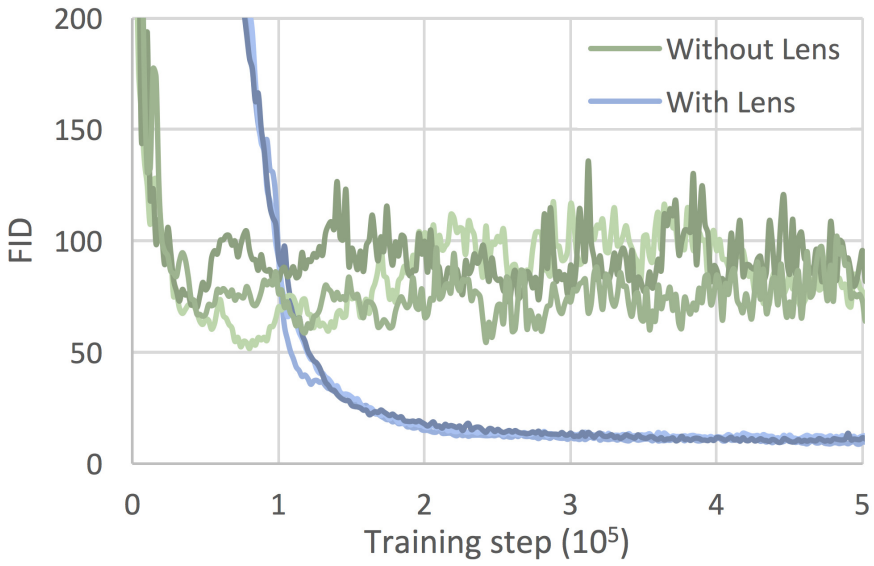


Figure 4.6: FID for DCGAN trained on the Color MNIST dataset. For each method, 3 independent runs with different random seeds for the weight initialization are shown. Since the value of λ is high early on during training, the GAN with a lens initially performs worse, but the quality soon catches up and surpasses that of the GAN without a lens as λ is lowered to a value of 0. The GANs with a lens are much more stable and more robust against different random seeds for the weight initialization.

during training, the process converges much more quickly and easily surpasses the quality of the GAN without a lens, yielding FID scores of 22 (with lens) vs. 42 (without lens). Additional experiments with much larger, heavily fine-tuned architectures that already show stable training for GANs did not show better FID after the addition of the lens, indicating that the proposed method can stabilize weaker architectures and lead to more robust GAN training with respect to hyperparameters.

Color MNIST

Since MNIST only has 10 main modes, it is not an adequate test for the mode collapse problem in GANs. To alleviate this, a color MNIST variant has been proposed (Srivastava *et al.*, 2017). Each sample is created by stacking three randomly drawn MNIST digits into the red, green and blue channels of an RGB image which leads to a dataset with 1000 modes (assuming 10 modes for MNIST) while still being easy to analyze visually.

As can be seen in Figure 4.7, the GAN without a lens first produces decent results in all color channels before it collapses partially. At this point, only the green color channel looks like MNIST digits while the other two channels are clearly not from the correct distribution. The FID reflects this, sometimes even increasing as training proceeds, with

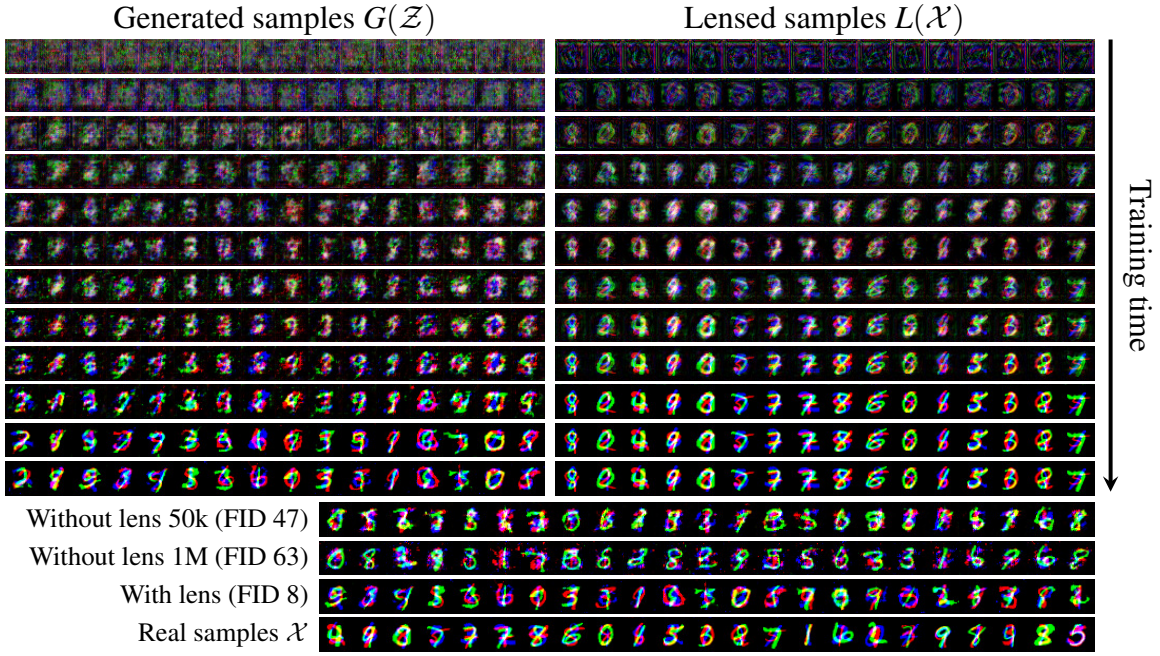


Figure 4.7: Results of DCGAN with a lens L on the Color MNIST dataset (top). The lens gradually improves reconstruction as G produces better samples. Once L is a perfect identity function, G adds remaining details and finally produces realistic results (bottom, third row). In comparison, the GAN without a lens only manages to produce good-looking digits in the green color channel and produces noise in the red and blue channels (bottom, first row, t=50k). As G improves quality in the green channel, the quality in the other two channels decreases (bottom, second row, t=1M) which is a commonly encountered instability during GAN training. Several runs with different random seeds for the weight initialization yielded similar results for both architectures, see Figure 4.6. Images best viewed in color.

values throughout training never getting lower than 50. Adding the lens to the GAN stabilizes training and leads to much higher quality samples with an FID of 9 for the best samples compared to 53 for the GAN without a lens.

4.4.2 LSGAN

MNIST

We found the LSGAN variant to be sensitive to the random seed for the weight initialization of the networks. LSGAN without a lens did not train in most cases, with the best run yielding FID scores of 19. With the lens, the networks always trained well, with the worst run producing FID scores of 16 and the best run giving FID scores of 14.

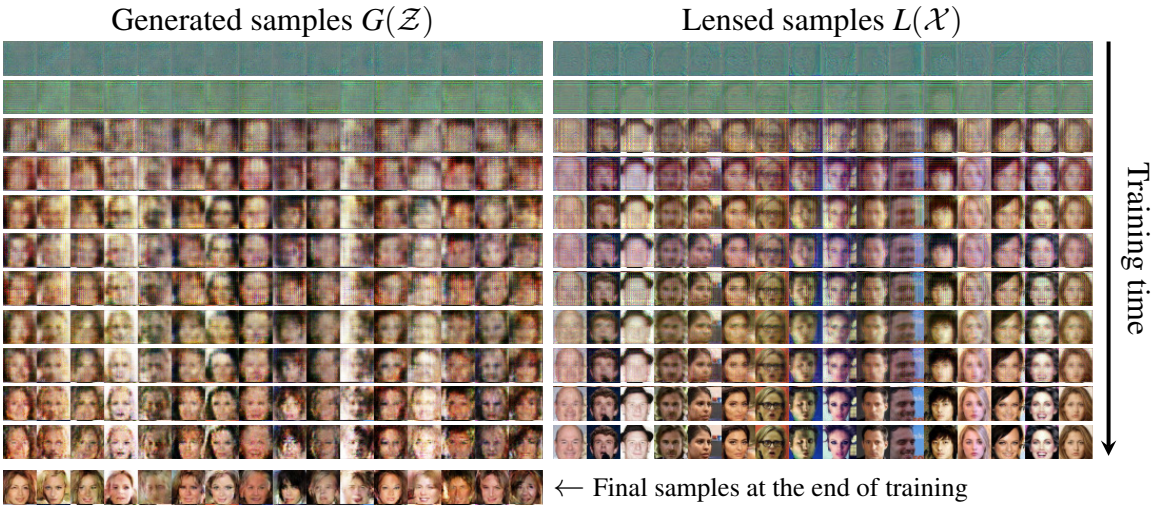


Figure 4.8: Generated and lensed samples at various steps during the training process of LSGAN on the CelebA dataset with a lens. The generator produces a large variety of faces since it is not forced to reproduce fine details early during training, making it less prone to the mode collapse problem.

Color MNIST

On the Color MNIST dataset, we found LSGAN to perform similarly. The best run without a lens yielded FID scores of 90 and training stalled there due to starved gradients. Adding the lens made the networks produce meaningful results in all runs, producing FID scores between 14 and 22 from different random initializations.

CelebA

On the CelebA dataset (Liu *et al.*, 2015a), LSGAN was unstable, with a starving generator early on during training due to a perfect discriminator that did not provide gradients. The best run without a lens yielded an FID score of 52. Adding the lens helped the system stabilize and produce meaningful results in all runs, with the best run yielding FID scores of 32 and the worst run yielding an FID of 37. Note that these numbers are comparably high due to the small model size of the generator and discriminator. The effects of the lens during training are shown in Figure 4.8.

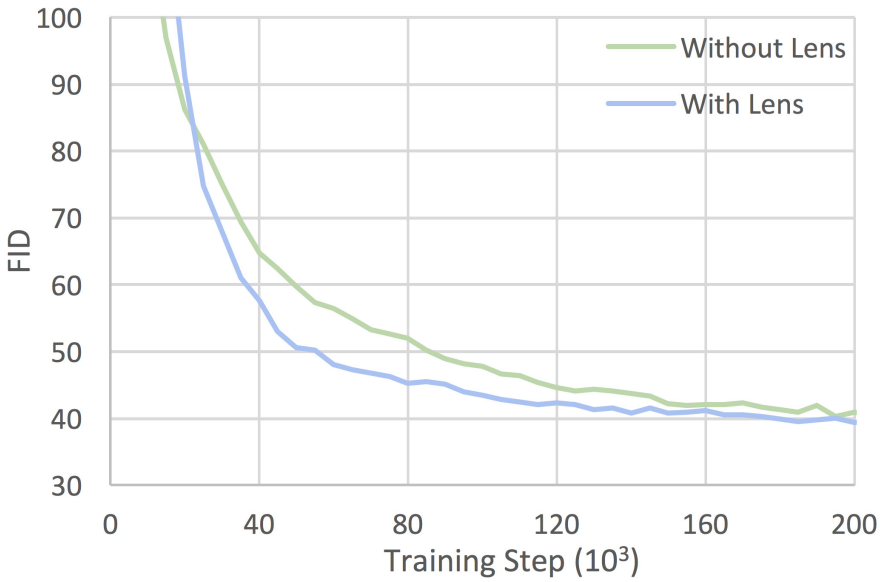


Figure 4.9: FID for WGAN-GP on Cifar-10 with and without a lens. The value for λ is smoothly lowered from 1 to 0 in the first K=10K steps. The final results have similar FIDs, but WGAN-GP with a lens converges faster to higher-quality samples. Tuning the rate at which λ is adapted could further improve convergence speeds.

4.4.3 WGAN-GP

Cifar-10

To test the lens on an entirely different GAN architecture, we also add it to the WGAN-GP framework (Gulrajani *et al.*, 2017). Wasserstein GANs are generally believed to be more stable than other GAN variants, making it harder for tweaks to significantly improve sample quality. Nevertheless, our experiment on the Cifar-10 dataset shows that the same lens with the same hyperparameters also works well with WGAN-GP, yielding higher-quality results as measured by the FID score at an earlier training stage. As seen in Figure 4.9, the model with a lens quickly surpasses the quality of the model without a lens and it takes some more training time for the GAN without a lens to catch up. When trained long enough, both models yield an FID of 39.

It is noteworthy that adding the lens can lead to faster training although the generator and discriminator are initially trained on a data distribution $L(\mathcal{X})$ that is quite different from the real data distribution $\mathcal{X}$ (see Figure 4.10). This result suggests that a scheduled learning procedure can indeed accelerate optimization of neural networks. The proposed lens is a natural way to dynamically adjust the rate at which learning proceeds.

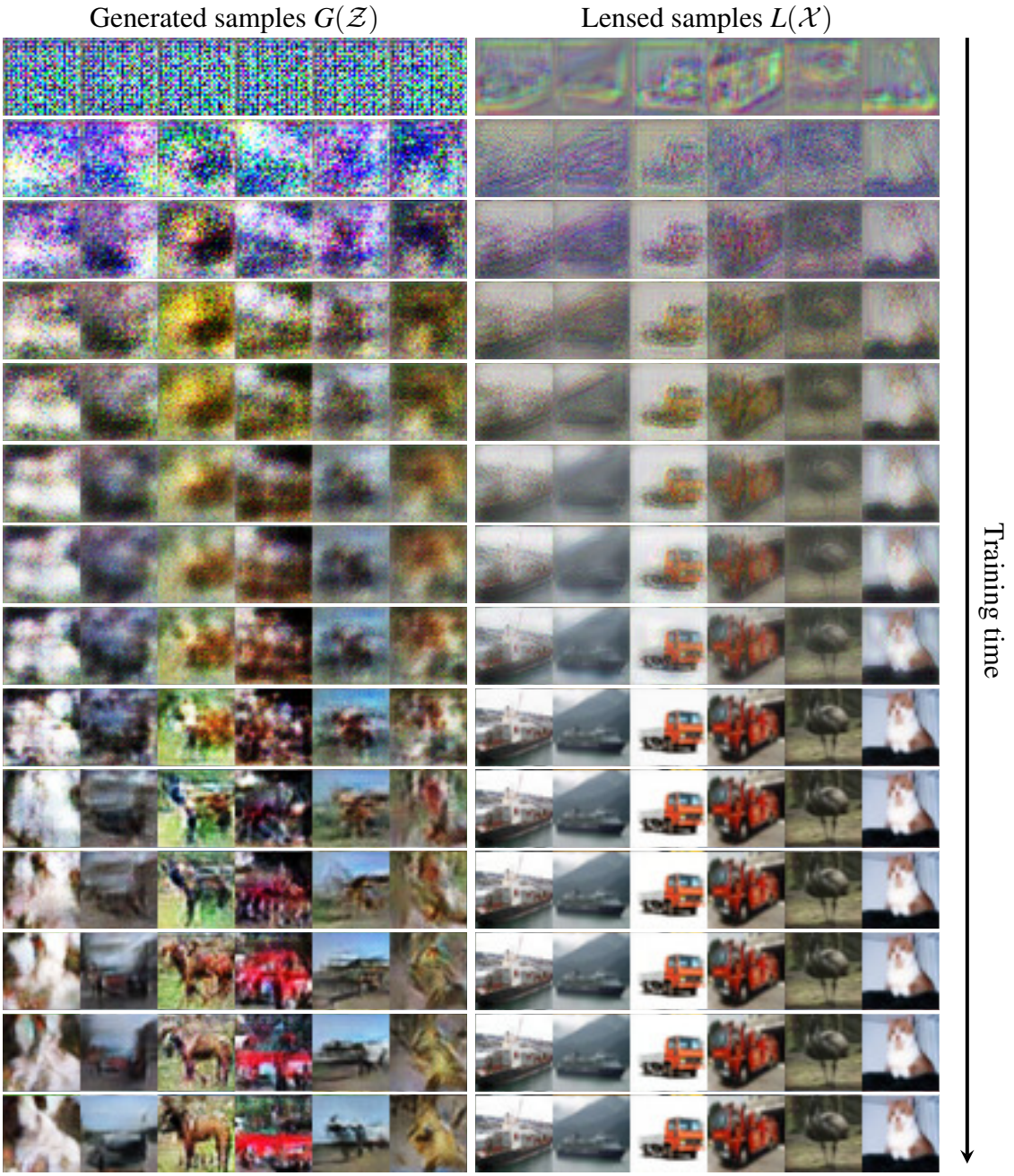


Figure 4.10: WGAN-GP with a lens L. In early training stages, the images are blurry lack contrast, but L gradually reconstructs finer details as G catches up. Note that by design, L could easily converge to the perfect identity mapping very quickly, so the gradual improvements seen here are a result of the adversarial loss term $\mathcal{L}_L^A$ rather than slow convergence.

4.5 Summary

We propose a generic module that leads to a dynamically self-adjusting progressive learning procedure of the target data distribution in GANs. A number of experiments on several GAN variants highlight the potential of this approach. Whilst the method is conceptually simple, it may have significant potential, not only in the image domain, but also in other domains such as audio or video generation. We hypothesize that similar modifications can be applied to improve optimization of other neural network architectures. For instance, autoencoders can be tempered by initially training to reconstruct lensed inputs, and recognition networks can be tempered by grouping or smoothing classes. Finally, it may be possible to incorporate prior knowledge about the task at hand by suitably biasing or initializing lenses, for instance using blurring lenses to generate images starting from low-frequency approximations.

Chapter 5

Assessing Generative Models via Precision and Recall

Recent advances in generative modeling have led to an increased interest in the study of statistical divergences as means of model comparison. Commonly used evaluation methods, such as the Fréchet Inception Distance (FID), correlate well with the perceived quality of samples and are sensitive to mode dropping. However, these metrics are unable to distinguish between different failure cases since they only yield one-dimensional scores. We propose a novel definition of precision and recall for distributions which disentangles the divergence into two separate dimensions. The proposed notion is intuitive, retains desirable properties, and naturally leads to an efficient algorithm that can be used to evaluate generative models. We relate this notion to total variation as well as to recent evaluation metrics such as Inception Score and FID. To demonstrate the practical utility of the proposed approach we perform an empirical study on several variants of Generative Adversarial Networks and Variational Autoencoders. In an extensive set of experiments we show that the proposed metric is able to disentangle the quality of generated samples from the coverage of the target distribution.

5.1 Introduction

Deep generative models, such as Variational Autoencoders (VAE) (Kingma and Welling, 2014) and Generative Adversarial Networks (GAN) (Goodfellow *et al.*, 2014), have received a great deal of attention due to their ability to learn complex, high-dimensional distributions. One of the biggest impediments to future research is the lack of quantitative evaluation methods to accurately assess the quality of trained models. Without a proper evaluation metric researchers often need to visually inspect generated samples or resort to qualitative techniques which can be subjective. One of the main difficulties for quantitative assessment lies in the fact that the distribution is only specified implicitly – one can learn to sample from a predefined distribution, but cannot evaluate the likelihood efficiently. In fact, even if likelihood computation were computationally tractable, it might be inadequate and misleading for high-dimensional problems (Theis *et al.*, 2016).

As a result, surrogate metrics are often used to assess the quality of the trained models. Some proposed measures, such as Inception Score (IS) (Salimans *et al.*, 2016b) and Fréchet Inception Distance (FID) (Heusel *et al.*, 2017), have shown promising results in practice. In particular, FID has been shown to be robust to image corruption, it correlates well with the visual fidelity of the samples, and it can be computed on unlabeled data.

However, all of the metrics commonly applied to evaluating generative models share a crucial weakness: Since they yield a one-dimensional score, they are unable to distinguish between different failure cases. For example, the generative models shown in Figure 5.1 obtain similar FIDs but exhibit different sample characteristics: the model on the left trained on MNIST (LeCun *et al.*, 1998) produces realistic samples, but only generates a subset of the digits. On the other hand, the model on the right produces low-quality samples which appear to cover all digits. A similar effect can be observed on the CelebA (Liu *et al.*, 2015b) dataset. In this work we argue that a single-value summary is not adequate to compare generative models.

Motivated by this shortcoming, we present a novel approach which disentangles the divergence between distributions into two components: *precision* and *recall*. Given a reference distribution P and a learned distribution Q, precision intuitively measures the quality of samples from Q, while recall measures the proportion of P that is covered by Q. Furthermore, we propose an elegant algorithm which can compute these quantities based on samples from P and Q. In particular, using this approach we are able to quantify the degree of *mode dropping* and *mode inventing* based on samples from the true and the learned distributions.

Our contributions: **(1)** We introduce a novel definition of precision and recall for distributions and prove that the notion is theoretically sound and has desirable properties, **(2)** we propose an efficient algorithm to compute these quantities, **(3)** we relate these notions to total variation, IS and FID, **(4)** we demonstrate that in practice one can quantify the degree of mode dropping and mode inventing on real world datasets (image and text data), and **(5)** we compare several types of generative models based on the proposed approach – to our knowledge, this is the first metric that experimentally confirms the folklore that GANs often produce ”sharper” images, but can suffer from mode collapse (high precision, low recall), while VAEs produce ”blurry” images, but cover more modes of the distribution (low precision, high recall).

5.2 Background and Related Work

The task of evaluating generative models is an active research area. Here we focus on recent work in the context of deep generative models for image and text data. Classic approaches relying on comparing log-likelihood have received some criticism due the fact that one can achieve high likelihood, but low image quality, and conversely, high-quality images but low likelihood (Theis *et al.*, 2016). While the likelihood can be approximated in some settings, kernel density estimation in high-dimensional spaces is

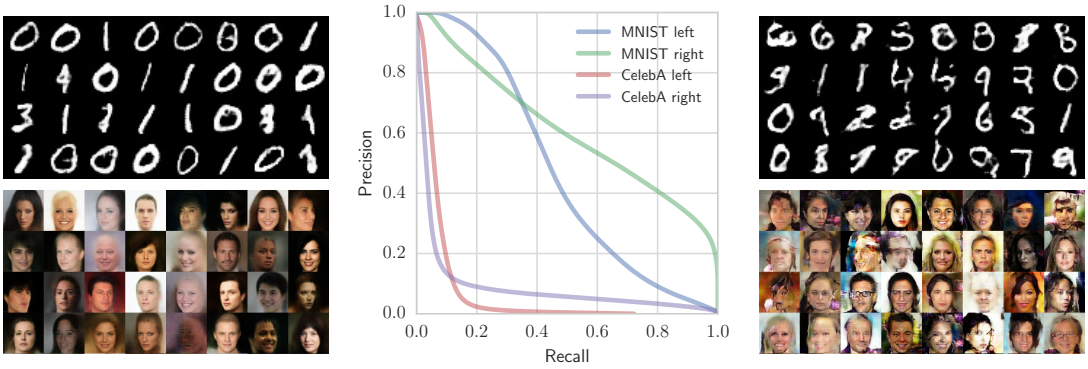


Figure 5.1: Comparison of GANs trained on MNIST and CelebA. Although the models obtain a similar FID on each dataset (32/29 for MNIST and 65/62 for CelebA), their samples look very different. For example, the model on the left produces reasonably looking faces on CelebA, but too many dark images. In contrast, the model on the right produces more artifacts, but more varied images. By the proposed metric (middle), the models on the left achieve higher precision and lower recall than the models on the right, which suffices to successfully distinguishing between the failure cases.

extremely challenging (Theis *et al.*, 2016; Wu *et al.*, 2017). Other failure modes related to density estimation in high-dimensional spaces have been elaborated in Huszár (2015); Theis *et al.* (2016). A recent review of popular approaches is presented in Borji (2018).

The Inception Score (IS) (Salimans *et al.*, 2016b) offers a way to quantitatively evaluate the quality of generated samples. Intuitively, the conditional label distribution $p(y|x)$ of samples containing meaningful objects should have low entropy, while the label distribution over the whole dataset $p(y)$ should have high entropy. The IS is formally defined as

$$\mathrm{IS}(G) = \exp(\mathbb{E}_{x\sim G}[d_{KL}(p(y|x), p(y)]). \tag{5.1}$$

The score is computed based on a classifier (Inception network trained on ImageNet). IS necessitates a labeled dataset and has been found to be weak at providing guidance for model comparison (Barratt and Sharma, 2018).

The FID (Heusel *et al.*, 2017) provides an alternative approach which requires no labeled data. The samples are first embedded in some feature space (*e.g.*, a specific layer of Inception network for images). Then, a continuous multivariate Gaussian is fit to the data and the distance computed as

$$\mathrm{FID}(x, g) = ||\mu_x - \mu_g||_2^2 \; + \; \mathrm{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x\Sigma_g)^{\frac{1}{2}}), \tag{5.2}$$

where μ and Σ denote the mean and covariance of the corresponding samples. FID is sensitive to both the addition of spurious modes as well as to mode dropping (see

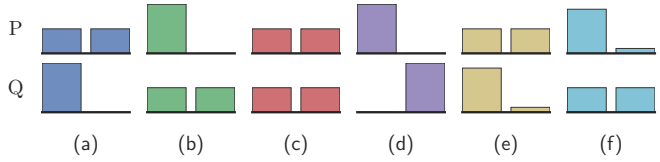


Figure 5.2: Intuitive examples of P and Q.

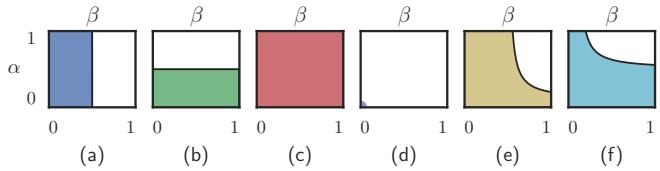


Figure 5.3: PRD for the examples above.

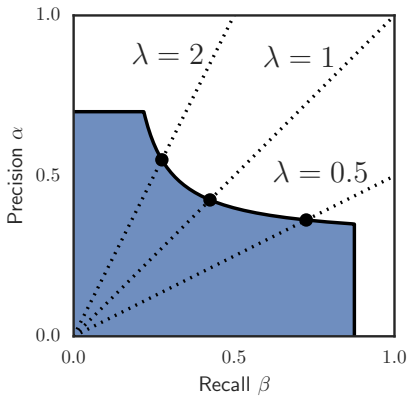


Figure 5.4: Illustration of the PRD algorithm.

Figure 5.5 and results in Lucic *et al.* (2018)). Bikowski *et al.* (2018) recently introduced an unbiased alternative to FID, the *Kernel Inception Distance*. While unbiased, it shares a very high Spearman rank-order correlation with FID (Kurach *et al.*, 2019).

Another approach is to train a classifier between the real and fake distributions and to use its accuracy on a test set as a proxy for the quality of the samples (Lopez-Paz and Oquab, 2016; Im *et al.*, 2018). This approach necessitates training of a classifier for each model which is seldom practical. Furthermore, the classifier might detect a single dimension where the true and generated samples differ (*e.g.*, barely visible artifacts in generated images) and enjoy high accuracy, which runs the risk of assigning lower quality to a better model.

To the best of our knowledge, all commonly used metrics for evaluating generative models are one-dimensional in that they only yield a single score or distance. A notion of precision and recall has previously been introduced by Lucic *et al.* (2018) where the authors compute the distance to the manifold of the true data and use it as a proxy for precision and recall on a synthetic dataset. Unfortunately, it is not possible to compute this quantity for more complex datasets.

5.3 PRD: Precision and Recall for Distributions

In this section, we derive a novel notion of precision and recall to compare a distribution Q to a reference distribution P. The key intuition is that *precision* should measure how much of Q can be generated by a "part" of P while *recall* should measure how much of P can be generated by a "part" of Q. Figure 5.2 (a)–(d) show four toy examples for P and Q to visualize this idea: (a) If P is bimodal and Q only captures one of the modes, we should have perfect precision but only limited recall. (b) In the opposite case, we should have perfect recall but only limited precision. (c) If $Q = P$, we should have perfect precision

and recall. (d) If the supports of P and Q are disjoint, we should have zero precision and recall. The examples (e) and (f) show the need for a tradeoff between precision and recall which we will discuss in Section 5.3.2.

5.3.1 Derivation

Let $S = \text{supp}(P) \cap \text{supp}(Q)$ be the (non-empty) intersection of the supports[1] of P and Q. Then, P may be viewed as a two-component mixture where the first component P_S is a probability distribution on S and the second component $P_{\overline{S}}$ is defined on the complement of S. Similarly, Q may be rewritten as a mixture of Q_S and $Q_{\overline{S}}$. More formally, for some $\bar{\alpha}, \bar{\beta} \in (0,1]$, we define

$$P = \bar{\beta} P_S + (1-\bar{\beta}) P_{\overline{S}} \quad \text{and} \quad Q = \bar{\alpha} Q_S + (1-\bar{\alpha}) Q_{\overline{S}}. \tag{5.3}$$

This decomposition allows for a natural interpretation: $P_{\overline{S}}$ is the part of P that cannot be generated by Q, so its mixture weight $1-\bar{\beta}$ may be viewed as a loss in recall. Similarly, $Q_{\overline{S}}$ is the part of Q that cannot be generated by P, so $1-\bar{\alpha}$ may be regarded as a loss in precision. In the case where $P_S = Q_S$, *i.e.*, the distributions P and Q agree on S up to scaling, $\bar{\alpha}$ and $\bar{\beta}$ provide us with a simple two-number precision and recall summary satisfying the examples in Figure 5.2 (a)–(d).

If $P_S \neq Q_S$, we are faced with a conundrum: Should the differences in P_S and Q_S be attributed to losses in precision or recall? Is Q_S inadequately "covering" P_S or is it generating "unnecessary" noise? Inspired by PR curves for binary classification, we propose to resolve this predicament by providing a trade-off between precision and recall instead of a two-number summary for any two distributions P and Q. To parametrize this trade-off, we consider a distribution μ on S that signifies a "true" common component of P_S and Q_S and similarly to (5.3), we decompose both P_S and Q_S as

$$P_S = \beta' \mu + (1-\beta') P_\mu \quad \text{and} \quad Q_S = \alpha' \mu + (1-\alpha') Q_\mu. \tag{5.4}$$

The distribution P_S is viewed as a two-component mixture where the first component is μ and the second component P_μ signifies the part of P_S that is "missed" by Q_S and should thus be considered a recall loss. Similarly, Q_S is decomposed into μ and the part Q_μ that signifies noise and should thus be considered a precision loss. As μ is varied, this leads to a trade-off between precision and recall.

It should be noted that unlike PR curves for binary classification where different thresholds lead to different classifiers, trade-offs between precision and recall here do not constitute different models or distributions – the proposed PRD curves only serve as a description of the characteristics of the model with respect to the target distribution.

[1] For a distribution P defined on a finite state space Ω, we define $\text{supp}(P) = \{\omega \in \Omega \mid P(\omega) > 0\}$.

5.3.2 Formal Definition

For simplicity, we consider distributions P and Q defined on a finite state space, though the notion of precision and recall can be extended to arbitrary distributions. By combining (5.3) and (5.4), we obtain the following formal definition of precision and recall.

Definition 1. *For $\alpha, \beta \in (0,1]$, the probability distribution Q has precision α at recall β w.r.t. P if there exist distributions μ, ν_P and ν_Q such that*

$$P = \beta\mu + (1-\beta)\nu_P \quad \textit{and} \quad Q = \alpha\mu + (1-\alpha)\nu_Q. \tag{5.5}$$

The component ν_P denotes the part of P that is "missed" by Q and encompasses both $P_{\overline{S}}$ in (5.3) and P_μ in (5.4). Similarly, ν_Q denotes the noise part of Q and includes both $Q_{\overline{S}}$ in (5.3) and Q_μ in (5.4).

Definition 2. *The set of attainable pairs of precision and recall of a distribution Q w.r.t. a distribution P is denoted by* $\mathrm{PRD}(Q,P)$ *and it consists of all (α,β) satisfying Definition 1 and the pair $(0,0)$.*

The set $\mathrm{PRD}(Q,P)$ characterizes the above-mentioned trade-off between precision and recall and can be visualized similarly to PR curves in binary classification: Figure 5.3 (a)–(d) show the set $\mathrm{PRD}(Q,P)$ on a 2D-plot for the examples (a)–(d) in Figure 5.2. Note how the plot distinguishes between (a) and (b): Any symmetric evaluation method (such as FID) assigns these cases the same score although they are highly different. The interpretation of the set $\mathrm{PRD}(Q,P)$ is further aided by the following set of basic properties.

Theorem 1. *Let P and Q be probability distributions defined on a finite state space Ω. The set* $\mathrm{PRD}(Q,P)$ *satisfies the following properties:*

(i) $Q = P \iff (1,1) \in \mathrm{PRD}(Q,P)$ *(equality)*

(ii) $\mathrm{supp}(Q) \cap \mathrm{supp}(P) = \emptyset \iff \mathrm{PRD}(Q,P) = \{(0,0)\}$ *(disjoint supports)*

(iii) $Q(\mathrm{supp}(P)) = \bar{\alpha} = \max_{(\alpha,\beta)\in\mathrm{PRD}(Q,P)} \alpha$ *(max precision)*

(iv) $P(\mathrm{supp}(Q)) = \bar{\beta} = \max_{(\alpha,\beta)\in\mathrm{PRD}(Q,P)} \beta$ *(max recall)*

(v) $\alpha' \in (0,\alpha]$, $\beta' \in (0,\beta]$, $(\alpha,\beta) \in \mathrm{PRD}(Q,P)$ $\implies (\alpha',\beta') \in \mathrm{PRD}(Q,P)$ *(monotonicity)*

(vi) $(\alpha,\beta) \in \mathrm{PRD}(Q,P) \iff (\beta,\alpha) \in \mathrm{PRD}(P,Q)$ *(duality)*

Property (i) in combination with Property (v) guarantees that $Q = P$ if and only if the set $\mathrm{PRD}(Q,P)$ contains the interior of the unit square, see case (c) in Figures 5.2 and 5.3. Similarly, Property (ii) assures that whenever there is no overlap between P and Q, $\mathrm{PRD}(Q,P)$ only contains the origin, see case (d) of Figures 5.2 and 5.3. Properties (iii)

and (iv) provide a connection to the decomposition in (5.3) and allow an analysis of the cases (a) and (b) in Figures 5.2 and 5.3: As expected, Q in (a) achieves a maximum precision of 1 but only a maximum recall of 0.5 while in (b), maximum recall is 1 but maximum precision is 0.5. Note that the quantities $\bar{\alpha}$ and $\bar{\beta}$ here are by construction the same as in (5.3). Finally, Property (vi) provides a natural interpretation of precision and recall: The precision of Q w.r.t. P is equal to the recall of P w.r.t. Q and *vice versa*.

Clearly, not all cases are as simple as the examples (a)–(d) in Figures 5.2 and 5.3, in particular if P and Q are different on the intersection S of their support. The examples (e) and (f) in Figure 5.2 and the resulting sets $\mathrm{PRD}(Q,P)$ in Figure 5.3 illustrate the importance of the trade-off between precision and recall as well as the utility of the set $\mathrm{PRD}(Q,P)$. In both cases, P and Q have the same support while Q has high precision and low recall in case (e) and low precision and high recall in case (f). This is clearly captured by the sets $\mathrm{PRD}(Q,P)$. Intuitively, the examples (e) and (f) may be viewed as noisy versions of the cases (a) and (b) in Figure 5.2.

We first show the following auxiliary result before proving Theorem 1.

Lemma 1. *Let P and Q be probability distributions defined on a finite state space Ω. Let $\alpha \in (0,1]$ and $\beta \in (0,1]$. Then, $(\alpha,\beta) \in \mathrm{PRD}(Q,P)$ if and only if there exists a distribution μ such that for all $\omega \in \Omega$*

$$P(\omega) \geq \beta\mu(\omega) \quad \textit{and} \quad Q(\omega) \geq \alpha\mu(\omega). \tag{5.6}$$

Proof. If $(\alpha,\beta) \in \mathrm{PRD}(Q,P)$, then (5.5) and the non-negativity of ν_P and ν_Q directly imply (5.6) for the same choice of μ. Conversely, if (5.6) holds for a distribution μ, we may define the distributions

$$\nu_P(\omega) = \frac{P(\omega) - \beta\mu(\omega)}{1-\beta} \quad \text{and} \quad \nu_Q(\omega) = \frac{Q(\omega) - \alpha\mu(\omega)}{1-\alpha}. \tag{5.7}$$

By definition α, β, μ, ν_P and ν_Q satisfy (5.5) in Definition 1, *i.e.* $(\alpha,\beta) \in \mathrm{PRD}(Q,P)$. □

Proof of Theorem 1. We show each of the properties independently.

(i) If $(1,1) \in \mathrm{PRD}(Q,P)$, then we have by Definition 1 that $P = \mu$ and $Q = \mu$ which implies $P = Q$ as claimed. Conversely, if $P = Q$, Definition 1 is satisfied for $\alpha = \beta = 1$ by choosing $\mu = \nu_P = \nu_Q = P$. Hence, $(1,1) \in \mathrm{PRD}(Q,P)$ as claimed.

(ii) We show both directions of the claim by contraposition, *i.e.*, we show

$$\mathrm{supp}(P) \cap \mathrm{supp}(Q) \neq \emptyset \iff \mathrm{PRD}(Q,P) \supset \{(0,0)\}. \tag{5.8}$$

Consider an arbitrary $\omega \in \mathrm{supp}(P) \cap \mathrm{supp}(Q)$. Then, by definition we have $P(\omega) > 0$ and $Q(\omega) > 0$. Let μ be defined as the distribution with $\mu(\omega) = 1$ and $\mu(\omega') = 0$ for all $\omega' \in \Omega \setminus \{\omega\}$. Clearly, it holds that $P(\omega) \geq P(\omega)\mu(\omega)$ and $Q(\omega) \geq Q(\omega)\mu(\omega)$ for

all $\omega \in \Omega$. Hence, by Lemma 1, we have $(Q(\omega), P(\omega)) \in \mathrm{PRD}(Q,P)$ which implies that $\mathrm{PRD}(Q,P) \supset \{(0,0)\}$ as claimed. Conversely, $\mathrm{PRD}(Q,P) \supset \{(0,0)\}$ implies by Lemma 1 that there exist $\alpha \in (0,1]$ and $\beta \in (0,1]$ as well as a distribution μ satisfying (5.6). Let $\omega \in \mathrm{supp}(\mu)$ which implies $\mu(\omega) > 0$ and thus by (5.6) also $P(\omega) > 0$ and $Q(\omega) > 0$. Hence, ω is in both of the supports of P and Q, *i.e.* $\mathrm{supp}(P) \cap \mathrm{supp}(Q) \neq \emptyset$ as claimed.

(iii) If $(\alpha,\beta) \in \mathrm{PRD}(Q,P)$, then by Lemma 1 there exists a distribution μ such that for all $\omega \in \Omega$ we have $P(\omega) \geq \beta\mu(\omega)$ and $Q(\omega) \geq \alpha\mu(\omega)$. $P(\omega) \geq \beta\mu(\omega)$ implies $\mathrm{supp}(\mu) \subseteq \mathrm{supp}(P)$ and hence $\sum_{\omega \in \mathrm{supp}(P)} \mu(\omega) = 1$. Together with $Q(\omega) \geq \alpha\mu(\omega)$, this yields

$$Q(\mathrm{supp}(P)) = \sum_{\omega \in \mathrm{supp}(P)} Q(\omega) \;\geq\; \alpha \sum_{\omega \in \mathrm{supp}(P)} \mu(\omega) = \alpha \tag{5.9}$$

which implies $\alpha \leq Q(\mathrm{supp}(P))$ for all $(\alpha,\beta) \in \mathrm{PRD}(Q,P)$. To prove the claim, we next show that there exists $(\alpha,\beta) \in \mathrm{PRD}(Q,P)$ with $\alpha = Q(\mathrm{supp}(P))$.

Let $S = \mathrm{supp}(P) \cap \mathrm{supp}(Q)$. If $S = \emptyset$, then $\alpha = Q(\mathrm{supp}(P)) = 0$ and $(0,0) \in \mathrm{PRD}(Q,P)$ by Definition 2 as claimed. For the case $S \neq \emptyset$, let $\beta = \min_{\omega \in S} P(\omega)Q(S)/Q(\omega)$. By definition of S, we have $\beta > 0$. Furthermore, $\beta \leq P(S) \leq 1$ since $P(\omega)/P(S) \leq Q(\omega)/Q(S)$ for at least one $\omega \in S$. Consider the distribution μ where $\mu(\omega) = Q(\omega)/Q(S)$ for all $\omega \in S$ and $\mu(\omega) = 0$ for $\omega \in \Omega \setminus S$. By construction, μ satisfies (5.6) in Lemma 1 and hence $(\alpha,\beta) \in \mathrm{PRD}(Q,P)$ as claimed.

(iv) This follows directly from applying Property (vi) to Property (iii).

(v) If $(\alpha,\beta) \in \mathrm{PRD}(Q,P)$, then by Lemma 1 there exists a distribution μ such that for all $\omega \in \Omega$ we have that $P(\omega) \geq \beta\mu(\omega)$ and $Q(\omega) \geq \alpha\mu(\omega)$. For $\alpha' \in (0,\alpha]$ and $\beta' \in (0,\beta]$, it follows that $P(\omega) \geq \beta'\mu(\omega)$ and $Q(\omega) \geq \alpha'\mu(\omega)$ for all $\omega \in \Omega$. By Lemma 1 this implies $(\alpha',\beta') \in \mathrm{PRD}(Q,P)$ as claimed.

(vi) This follows directly from swapping α, P, ν_P with β, Q, ν_Q in Definition 1.

□

5.3.3 Algorithm

Computing the set $\mathrm{PRD}(Q,P)$ based on Definitions 1 and 2 is non-trivial as one has to check whether there exist suitable distributions μ, ν_P and ν_Q for all possible values of α and β. We introduce an equivalent definition of $\mathrm{PRD}(Q,P)$ in Theorem 2 that does not depend on the distributions μ, ν_P and ν_Q and that leads to an elegant algorithm to compute practical PRD curves.

Theorem 2. *Let P and Q be two probability distributions defined on a finite state space Ω. For $\lambda > 0$ define the functions*

$$\alpha(\lambda) = \sum_{\omega \in \Omega} \min\left(\lambda P(\omega), Q(\omega)\right) \quad \textit{and} \quad \beta(\lambda) = \sum_{\omega \in \Omega} \min\left(P(\omega), \frac{Q(\omega)}{\lambda}\right). \tag{5.10}$$

Then, it holds that

$$\mathrm{PRD}(Q,P) = \{(\theta\alpha(\lambda), \theta\beta(\lambda)) \mid \lambda \in (0,\infty), \theta \in [0,1]\}. \tag{5.11}$$

Proof. We first show that

$$\mathrm{PRD}(Q,P) \subseteq \{(\theta\alpha(\lambda), \theta\beta(\lambda)) \mid \lambda \in (0,\infty), \theta \in [0,1]\} \tag{5.12}$$

by considering any $(\alpha', \beta') \in \mathrm{PRD}(Q,P)$ and showing that $(\alpha', \beta') = (\theta\alpha(\lambda), \theta\beta(\lambda))$ for some $\lambda \in (0,\infty)$ and $\theta \in [0,1]$.

For the case $(\alpha', \beta') = (0,0)$, the result holds trivially for the choice of $\lambda = 1$ and $\theta = 0$. Otherwise, *i.e.* for $(\alpha', \beta') \neq (0,0)$, we choose $\lambda = \alpha'/\beta'$ and $\theta = \beta'/\beta(\lambda)$. Since $\alpha(\lambda) = \lambda\beta(\lambda)$ by definition, this implies $(\alpha', \beta') = (\theta\alpha(\lambda), \theta\beta(\lambda))$ as required. Furthermore, $\lambda \in (0,\infty)$ since by Definitions 1 and 2 $\alpha' > 0$ if and only if $\beta' > 0$. Similarly, we show that $\theta \in [0,1]$: By Lemma 1 there exists a distribution μ such that $\beta'\mu(\omega) \leq P(\omega)$ and $\alpha'\mu(\omega) \leq Q(\omega)$ for all $\omega \in \Omega$. This implies that $\beta'\mu(\omega) \leq Q(\omega)/\lambda$ and thus $\beta'\mu(\omega) \leq \min\left(P(\omega), Q(\omega)/\lambda\right)$ for all $\omega \in \Omega$. Summing over all $\omega \in \Omega$, we obtain

$$\beta' \leq \sum_{\omega \in \Omega} \min\left(P(\omega), \frac{Q(\omega)}{\lambda}\right) = \beta(\lambda) \tag{5.13}$$

which implies $\theta \in [0,1]$. Finally, we show that

$$\mathrm{PRD}(Q,P) \supseteq \{(\theta\alpha(\lambda), \theta\beta(\lambda)) \mid \lambda \in (0,\infty), \theta \in [0,1]\}. \tag{5.14}$$

Consider arbitrary $\lambda \in (0,\infty)$ and $\theta \in [0,1]$. If $\beta(\lambda) = 0$, the claim holds trivially since we have $(0,0) \in \mathrm{PRD}(Q,P)$. Otherwise, we can define the distribution μ such that

$$\mu(\omega) = \min\left(P(\omega), \frac{Q(\omega)}{\lambda}\right) / \beta(\lambda) \tag{5.15}$$

for all $\omega \in \Omega$. Then, by definition, we have

$$\beta(\lambda)\mu(\omega) \leq \min\left(P(\omega), \frac{Q(\omega)}{\lambda}\right) \leq P(\omega) \quad \text{for all } \omega \in \Omega, \tag{5.16}$$

and equivalently

$$\alpha(\lambda)\mu(\omega) \leq \min\left(\lambda P(\omega), Q(\omega)\right) \leq Q(\omega) \quad \text{for all } \omega \in \Omega \tag{5.17}$$

since $\alpha(\lambda) = \lambda\beta(\lambda)$. Because $\theta \in [0,1]$, this implies both $\theta\beta(\lambda)\mu(\omega) \leq P(\omega)$ and $\theta\alpha(\lambda)\mu(\omega) \leq Q(\omega)$ for all $\omega \in \Omega$. Hence, by Lemma 1, $(\theta\alpha(\lambda), \theta\beta(\lambda)) \in \mathrm{PRD}(Q,P)$ for all $\lambda \in (0,\infty)$ and $\theta \in [0,1]$ as claimed. □

The key idea of Theorem 2 is illustrated in Figure 5.4: The set $\mathrm{PRD}(Q,P)$ may be seen as a union of segments of lines $\alpha = \lambda\beta$ over all $\lambda \in (0,\infty)$. Each segment starts at the origin $(0,0)$ and ends at the maximal achievable value $(\alpha(\lambda), \beta(\lambda))$. This provides a surprisingly simple algorithm to compute $\mathrm{PRD}(Q,P)$ in practice: Simply compute pairs of $\alpha(\lambda)$ and $\beta(\lambda)$ as in (5.10) for an equiangular grid of values of λ. For a given angular resolution $m \in \mathbb{N}$, we compute

$$\widehat{\mathrm{PRD}}(Q,P) = \{(\alpha(\lambda), \beta(\lambda)) \mid \lambda \in \Lambda\} \quad \text{where} \quad \Lambda = \left\{\tan\left(\tfrac{i}{m+1}\tfrac{\pi}{2}\right) \mid i = 1,2,\ldots,m\right\}.$$

To compare different distributions Q_i, one may simply plot their respective PRD curves $\widehat{\mathrm{PRD}}(Q_i,P)$, while an approximation of the full sets $\mathrm{PRD}(Q_i,P)$ may be computed by interpolation between $\widehat{\mathrm{PRD}}(Q_i,P)$ and the origin. An implementation of the algorithm is available at github.com/msmsajjadi/precision-recall-distributions.

5.3.4 Connection to Total Variation Distance

Theorem 2 provides a natural interpretation of the proposed approach. For $\lambda = 1$, we have

$$\alpha(1) = \beta(1) = \sum_{\omega\in\Omega} \min\left(P(\omega), Q(\omega)\right) = \sum_{\omega\in\Omega} \left[P(\omega) - \left(P(\omega) - Q(\omega)\right)^+\right] = 1 - \delta(P,Q)$$

where $\delta(P,Q)$ denotes the total variation distance between P and Q. As such, our notion of precision and recall may be viewed as a generalization of total variation distance.

5.4 Application to Deep Generative Models

In this section, we show that the algorithm introduced in Section 5.3.3 can be readily applied to evaluate precision and recall of deep generative models. In practice, access to P and Q is given via samples $\hat{P} \sim P$ and $\hat{Q} \sim Q$. Given that both P and Q are continuous distributions, the probability of generating a point sampled from Q is 0. Furthermore, there is strong empirical evidence that comparing samples in image space runs the risk of assigning higher quality to a worse model (Lopez-Paz and Oquab, 2016; Salimans *et al.*, 2016b; Theis *et al.*, 2016). A common remedy is to apply a pre-trained classifier trained on natural images and to compare $\hat{P}$ and $\hat{Q}$ at a feature level. Intuitively, in this feature space the samples should be compared based on statistical regularities in the images rather than random artifacts resulting from the generative process (Lopez-Paz and Oquab, 2016; Odena *et al.*, 2016).

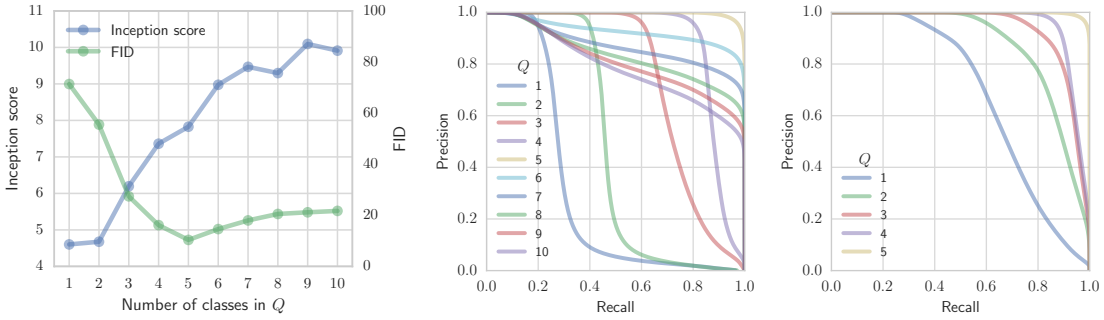


Figure 5.5: Left: IS and FID as we remove and add classes of CIFAR-10. IS generally only increases, while FID is sensitive to both the addition and removal of classes. However, it cannot distinguish between the two failure cases of inventing or dropping modes. Middle: Resulting PRD curves for the same experiment. As expected, adding modes leads to a loss in precision (Q_6–Q_{10}), while dropping modes leads to a loss in recall (Q_1–Q_4). As an example consider Q_4 and Q_6 which have similar FID, but strikingly different PRD curves. The same behavior can be observed for the task of text generation, as displayed on the plot on the right. For this experiment, we set P to contain samples from all classes so the PRD curves demonstrate the increase in recall as we increase the number of classes in Q.

Following this line of work, we first use a pre-trained Inception network to embed the samples (*i.e.*, using the *Pool3* layer (Heusel *et al.*, 2017)). We then cluster the union of $\hat{P}$ and $\hat{Q}$ in this feature space using mini-batch k-means with $k = 20$ (Sculley, 2010). Intuitively, we reduce the problem to a one dimensional problem where the histogram over the cluster assignments can be meaningfully compared. Hence, failing to produce samples from a cluster with many samples from the true distribution will hurt recall, and producing samples in clusters without many real samples will hurt precision. As the clustering algorithm is randomized, we run the procedure several times and average over the PRD curves. We note that such a clustering is meaningful as shown in Figure 5.9 and that it can be efficiently scaled to very large sample sizes (Bachem *et al.*, 2016).

We stress that from the point of view of the proposed algorithm, only a meaningful embedding is required. As such, the algorithm can be applied to various data modalities. In particular, we show in Section 5.4.1 that besides image data the algorithm can be applied to a text generation task.

5.4.1 Adding and Dropping Modes from the Target Distribution

Mode collapse or mode dropping is a major challenge in GANs (Goodfellow *et al.*, 2014; Salimans *et al.*, 2016b). Due to the symmetry of commonly used metrics with respect to precision and recall, the only way to assess whether the model is producing low-quality images or dropping modes is by visual inspection. In stark contrast, the proposed metric can quantitatively disentangle these effects which we empirically demonstrate.

We consider three datasets commonly used in the GAN literature: MNIST (LeCun *et al.*, 1998), Fashion-MNIST (Xiao *et al.*, 2017), and CIFAR-10 (Krizhevsky and Hinton, 2009). These datasets are labeled and consist of 10 balanced classes. To show the sensitivity of the proposed measure to mode dropping and mode inventing, we first fix $\hat{P}$ to contain samples from the first 5 classes in the respective test set. Then, for a fixed $i = 1, \ldots, 10$, we generate a set $\hat{Q}_i$, which consists of samples from the first i classes from the training set. As i increases, $\hat{Q}_i$ covers an increasing number of classes from $\hat{P}$ which should result in higher recall. As we increase i beyond 5, $\hat{Q}_i$ includes samples from an increasing number of classes that are not present in $\hat{P}$ which should result in a loss in precision, but not in recall as the other classes are already covered. Finally, the set $\hat{Q}_5$ covers the same classes as $\hat{P}$, so it should have high precision and high recall.

Figure 5.5 (left) shows the IS and FID for the CIFAR-10 dataset while results on further datasets are shown in Figure 5.12. Since the IS is not computed w.r.t. a reference distribution, it is invariant to the choice of $\hat{P}$, so as we add classes to $\hat{Q}_i$, the IS increases. The FID decreases as we add more classes until $\hat{Q}_5$ before it starts to increase as we add spurious modes. Critically, FID fails to distinguish the cases of mode dropping and mode inventing: $\hat{Q}_4$ and $\hat{Q}_6$ share similar FIDs. In contrast, Figure 5.5 (middle) shows our PRD curves as we vary the number of classes in $\hat{Q}_i$. Adding correct modes leads to an increase in recall, while adding fake modes leads to a loss of precision.

We also apply the proposed approach on text data as shown in Figure 5.5 (right). In particular, we use the MultiNLI corpus of crowd-sourced sentence pairs annotated with topic and textual entailment information (Williams *et al.*, 2018). After discarding the entailment label, we collect all unique sentences for the same topic. Following Cífka *et al.* (2018), we embed these sentences using a BiLSTM with 2048 cells in each direction and max pooling, leading to a 4096-dimensional embedding (Conneau *et al.*, 2017). We consider 5 classes from this dataset and fix $\hat{P}$ to contain samples from all classes to measure the loss in recall for different Q_i. The PRD curves in Figure 5.5 (right) successfully demonstrate the sensitivity of recall to mode dropping.

5.4.2 Assessing Class Imbalances for GANs

In this section we analyze the effect of class imbalance on the PRD curves. Figure 5.6 shows a pair of GANs trained on MNIST which have virtually the same FID, but very different PRD curves.

The model on the left generates a subset of the digits of high quality, while the model on the right seems to generate all digits, but each has low quality. We can naturally interpret this difference via the PRD curves: For a desired recall level of less than $\sim$0.6, the model on the left enjoys higher precision – it generates several digits of high quality. If, however, one desires a recall higher than $\sim$0.6, the model on the right enjoys higher precision as it covers all digits.

To confirm this, we train an MNIST classifier on the embedding of $\hat{P}$ with the ground truth labels and plot the distribution of the predicted classes for both models. The

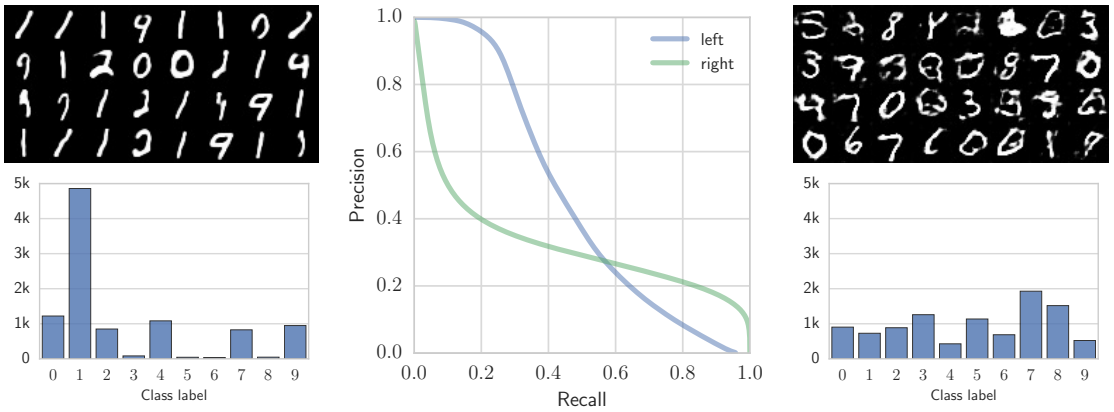


Figure 5.6: Comparing two GANs trained on MNIST which both achieve an FID of 49. The model on the left seems to produce high-quality samples of only a subset of digits. On the other hand, the model on the right generates low-quality samples of all digits. The histograms showing the corresponding class distributions based on a trained MNIST classifier confirm this observation. At the same time, the classifier is more confident which indicates different levels of precision (96.7% for the model on the left compared to 88.6% for the model on the right). Finally, we note that the proposed PRD algorithm does not require labeled data, as opposed to the IS which further needs a classifier that was trained on the respective dataset.

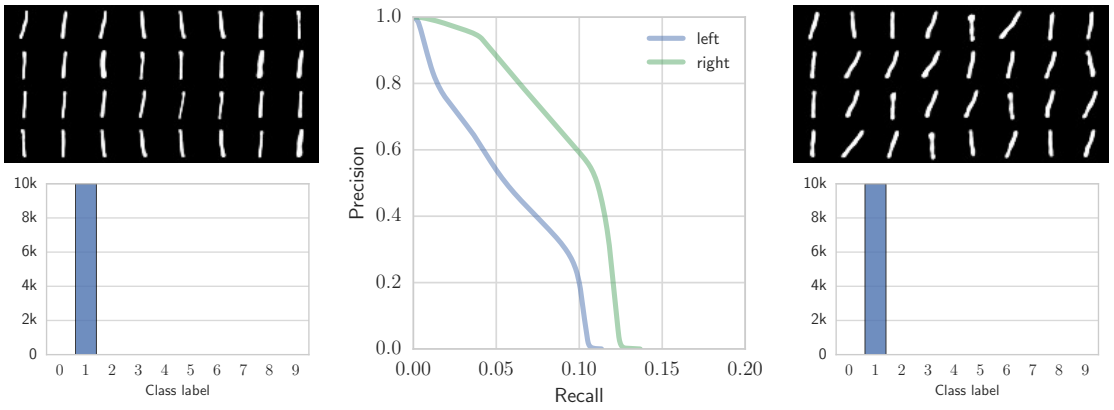


Figure 5.7: Comparing a pair of GANs on MNIST which have both collapsed to producing 1's. An analysis with a trained classifier as in Section 5.4.2 comes to the same conclusion for both models, namely, that they have collapsed to producing 1's only. However, the PRD curve correctly shows that the model on the right has a slightly higher recall: while the model on the left is producing straight 1's only, the model on the right is producing some more varied shapes such as tilted 1's. Note the cropped scale on the *Recall* axis.

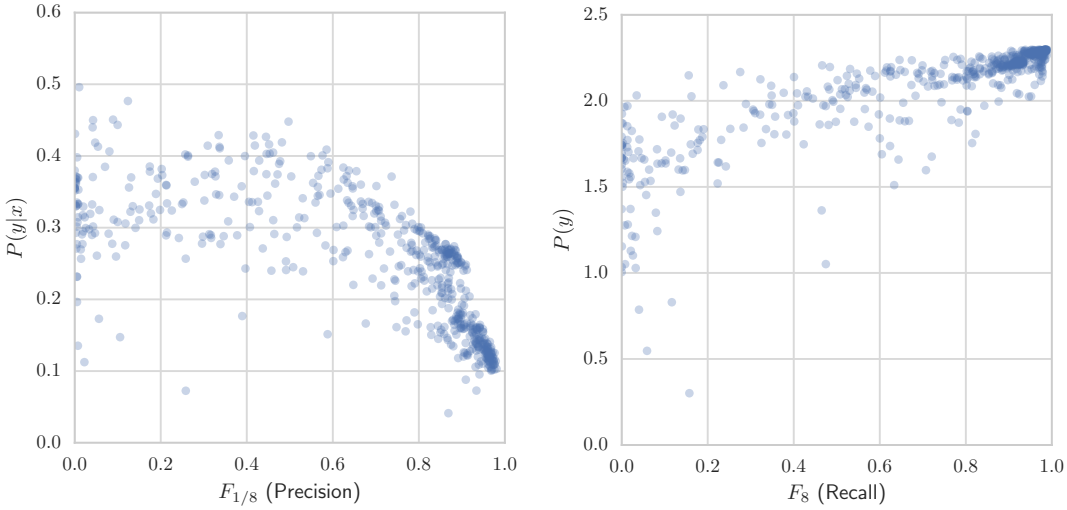


Figure 5.8: Comparing our unsupervised $F_{1/8}$ and F_8 measures with the supervised measures $P(y|x)$ and $P(y)$ similar to the IS (for a definition of F_β, see Section 5.4.3). Each circle represents a trained generative model (GAN or VAE) on the MNIST dataset. The values show a fairly high correlation with a Spearman rank correlation coefficient of -0.83 on the left and 0.89 on the right.

histograms clearly show that the model on the left failed to generate all classes (loss in recall), while the model on the right is producing a more balanced distribution over all classes (high recall). At the same time, the classifier has an average *confidence*[2] of 96.7% on the model on the left compared to 88.6% on the model on the right, indicating that the sample quality of the former is higher. This aligns very well with the PRD plots: samples on the left have high quality but are not diverse in contrast to the samples on the right which are diverse but have low quality.

This analysis reveals a connection to IS which is based on the premise that the conditional label distribution $p(y|x)$ should have low entropy, while the marginal $p(y) = \int p(y|x = G(z))dz$ should have high entropy. To further analyze the relationship between the proposed approach and PRD curves, we plot $p(y|x)$ against precision and $p(y)$ against recall in Figure 5.8. The results over a large number of GANs and VAEs show a large Spearman correlation of -0.83 for precision and 0.89 for recall. We however stress two key differences between the approaches: Firstly, to compute the quantities in IS one needs a classifier and a labeled dataset in contrast to the proposed PRD metric which can be applied on unlabeled data. Secondly, IS only captures losses in recall w.r.t. classes, while our metric measures more fine-grained recall losses, see Figure 5.7.

[2]We denote the output of the classifier for its highest value at the softmax layer as confidence. The intuition is that higher values signify higher confidence of the model for the given label.

Figure 5.9: Clustering the real and generated samples from a GAN in feature space (10 cluster centers for visualization) yields the clusters above for the datasets MNIST, Fashion-MNIST, CIFAR-10 and CelebA. Although the GAN samples are not perfect, they are clustered in a meaningful way.

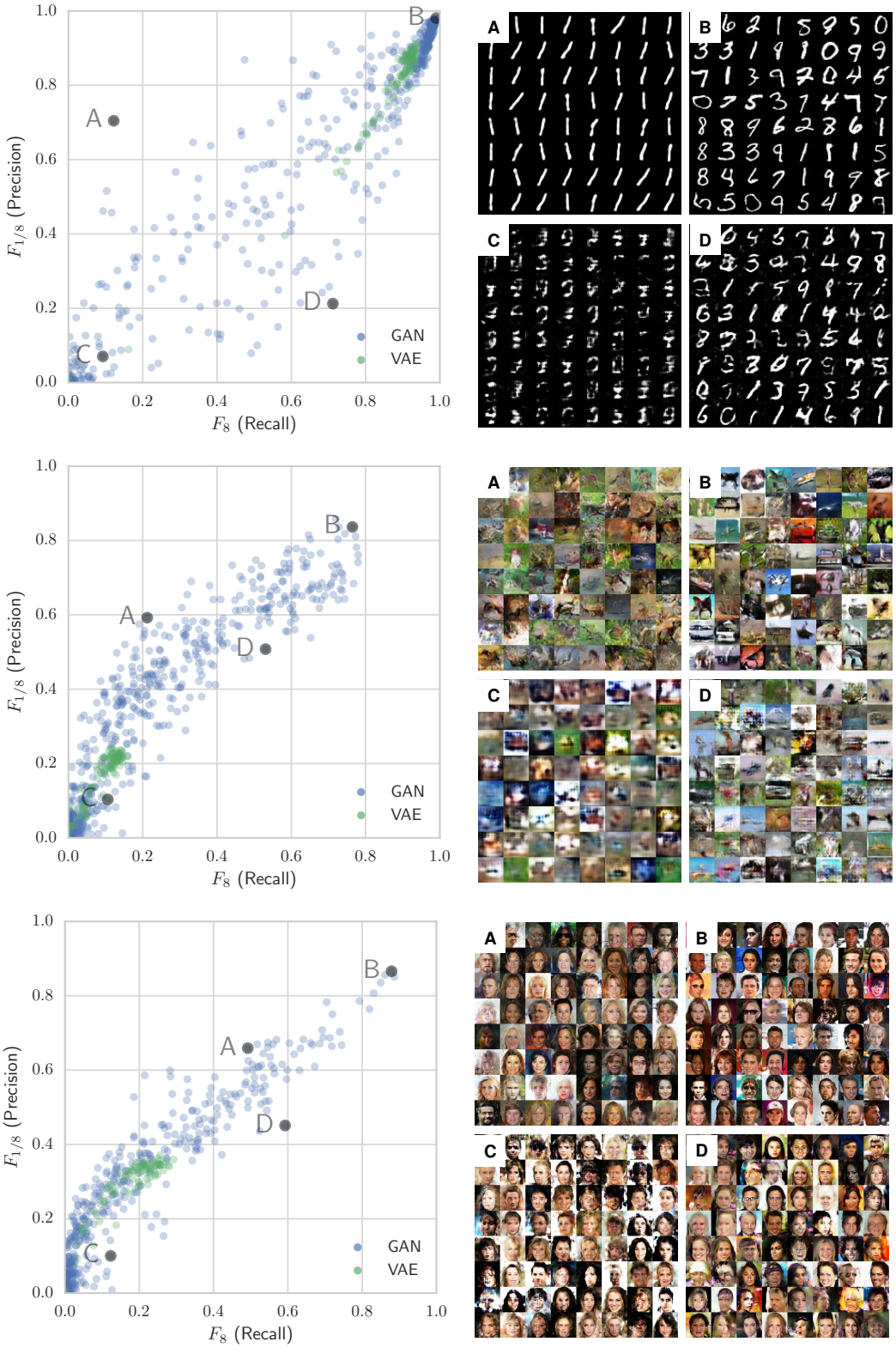


Figure 5.10: Corresponding plots as in Figure 5.11 for the datasets MNIST (top), CIFAR-10 (middle) and CelebA (bottom).

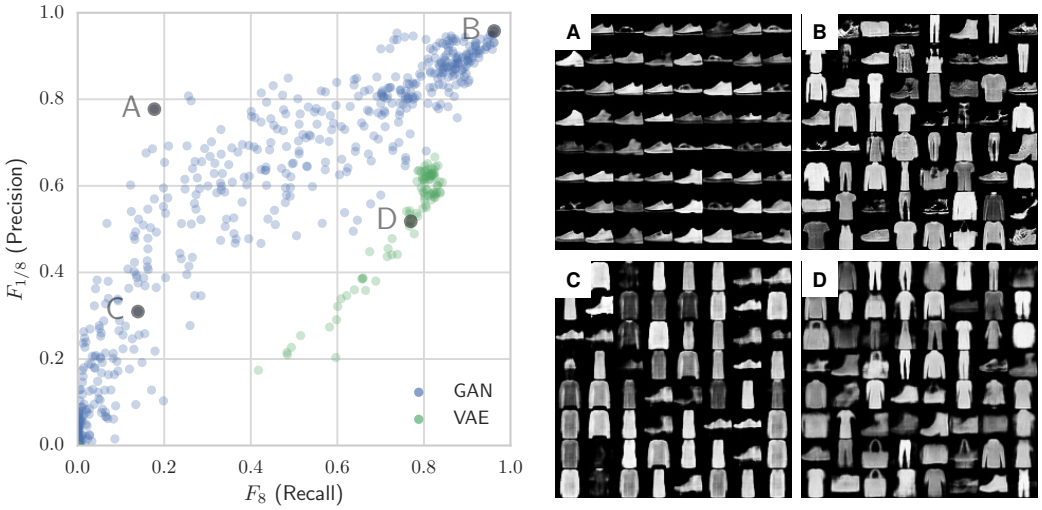


Figure 5.11: $F_{1/8}$ vs F_8 scores for a large number of GANs and VAEs on the Fashion-MNIST dataset. For each model, we plot the maximum $F_{1/8}$ and F_8 scores to show the trade-off between precision and recall. VAEs generally achieve lower precision and/or higher recall than GANs which matches the folklore that VAEs often produce samples of lower quality while being less prone to mode collapse. On the right we show samples from four models which correspond to various success/failure modes: (A) high precision, low recall, (B) high precision, high recall, (C) low precision, low recall, and (D) low precision, high recall.

5.4.3 Large-Scale Evaluation of 800 GANs and VAEs

We evaluate the precision and recall of 7 GAN types and the VAE with 100 hyperparameter settings each as provided by Lucic *et al.* (2018). In order to visualize this vast quantity of models, one needs to summarize the PRD curves. A natural idea is to compute the maximum F_1 score, which corresponds to the harmonic mean between precision and recall as a single-number summary. This idea is fundamentally flawed as F_1 is symmetric. However, its generalization, defined as

$$F_\beta = (1+\beta^2)\frac{p \cdot r}{(\beta^2 p)+r} \tag{5.18}$$

provides a way to quantify the relative importance of precision and recall: $\beta > 1$ weighs recall higher than precision, whereas $\beta < 1$ weighs precision higher than recall. As a result, we propose to distill each PRD curve into a pair of values: F_β and $F_{1/\beta}$.

Figure 5.11 compares the maximum F_8 with the maximum $F_{1/8}$ for these models on the Fashion-MNIST dataset. We choose $\beta = 8$ as it offers a good insight into the bias towards precision versus recall. Since F_8 weighs recall higher than precision and $F_{1/8}$

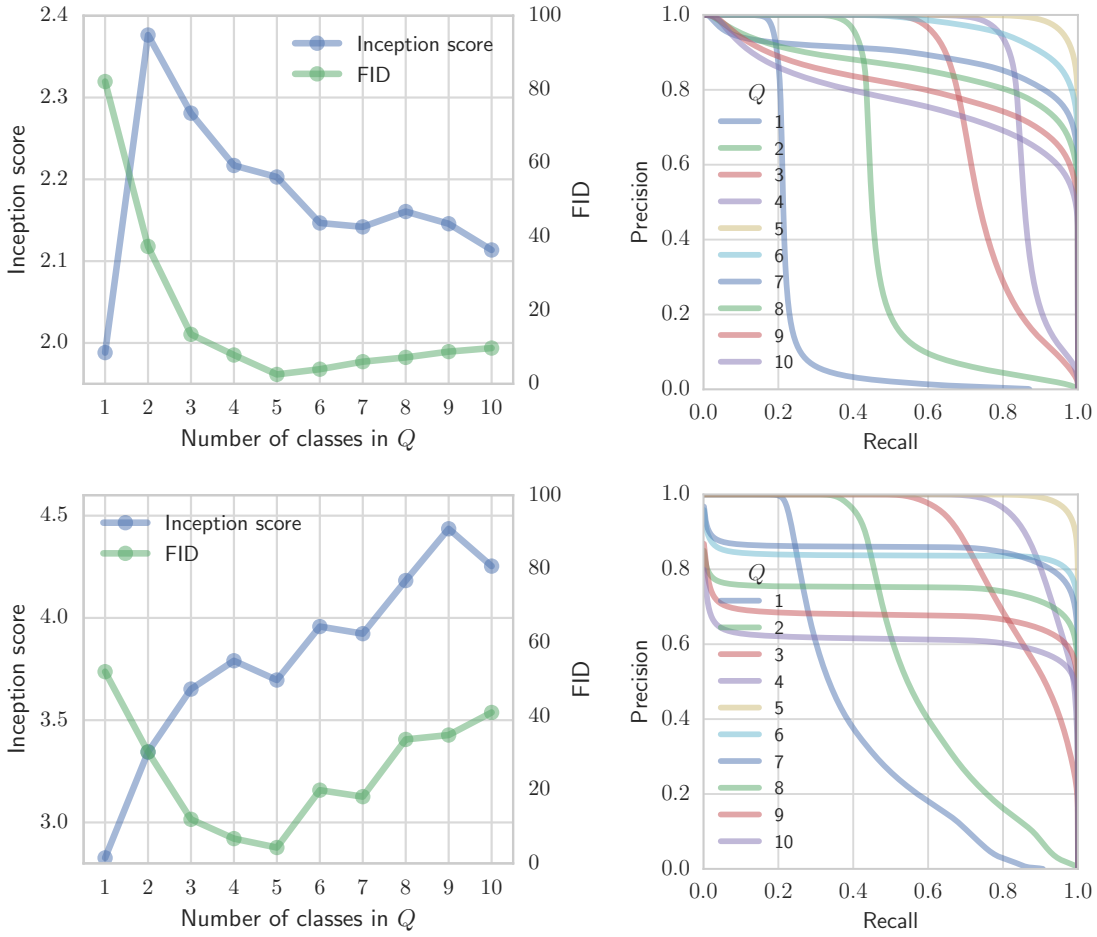


Figure 5.12: Corresponding plots as in Figure 5.5 for the datasets MNIST (top) and Fashion-MNIST (bottom).

does the opposite, models with higher recall than precision will lie below the diagonal $F_8 = F_{1/8}$ and models with higher precision than recall will lie above. To our knowledge, this is the first metric which confirms the folklore that VAEs are biased towards higher recall, but may suffer from precision issues (*e.g.*, due to blurring effects), at least on this dataset. On the right, we show samples from four models on the extreme ends of the plot for all combinations of high and low precision and recall. Figure 5.10 on page 78 shows similar graphics for the MNIST, CIFAR-10 and CelebA datasets.

5.5 Summary

Quantitatively evaluating generative models is a challenging task of paramount importance. In this work we show that one-dimensional scores are not sufficient to capture different

failure cases of current state-of-the-art generative models. As an alternative, we propose a novel notion of precision and recall for distributions and prove that both notions are theoretically sound and have desirable properties. We then connect these notions to total variation distance as well as FID and IS and we develop an efficient algorithm that can be readily applied to evaluate deep generative models based on samples. We investigate the properties of the proposed algorithm on real-world datasets, including image and text generation, and show that it captures the precision and recall of generative models. Finally, we find empirical evidence supporting the folklore that VAEs produce samples of lower quality, while being less prone to mode collapse than GANs.

Chapter 6

From Variational to Deterministic Autoencoders

Variational Autoencoders (VAEs) provide a theoretically-backed framework for deep generative models. However, they often produce "blurry" images, which is linked to their training objective. Sampling in the most popular implementation, the Gaussian VAE, can be interpreted as simply injecting noise to the input of a deterministic decoder. In practice, this simply enforces a smooth latent space structure. We challenge the adoption of the full VAE framework on this specific point in favor of a simpler, deterministic one. Specifically, we investigate how substituting stochasticity with other explicit and implicit regularization schemes can lead to a meaningful latent space without having to force it to conform to an arbitrarily chosen prior. To retrieve a generative mechanism for sampling new data points, we propose to employ an efficient ex-post density estimation step that can be readily adopted both for the proposed deterministic autoencoders as well as to improve sample quality of existing VAEs. We show in a rigorous empirical study that regularized deterministic autoencoding achieves state-of-the-art sample quality on the common MNIST, CIFAR-10 and CelebA datasets.

6.1 Introduction

Generative modeling lies at the core of machine learning and computer vision. By capturing the mechanisms behind the data generation process, one can reason about data probabilistically, access and traverse the low-dimensional manifold the data is assumed to live on, and ultimately *generate new data*. It is therefore not surprising that learning generative models has gained momentum in applications like chemistry (Jin *et al.*, 2018; Gómez-Bombarelli *et al.*, 2018), NLP (Severyn *et al.*, 2017; Bowman *et al.*, 2016) and computer vision (Brock *et al.*, 2019; Sohn *et al.*, 2015).

Variational Autoencoders (VAEs) (Kingma and Welling, 2014; Rezende *et al.*, 2014) allow for a principled probabilistic way to model high-dimensional distributions. They do so by casting learning representations as a variational inference problem. Learning a VAE amounts to the optimization of an objective balancing the quality of autoencoded samples

through a stochastic encoder–decoder pair while encouraging the latent space to follow a fixed prior distribution.

Since their introduction, VAEs have become one of the frameworks of choice for generative modeling, promising theoretically well-founded and more stable training than Generative Adversarial Networks (GANs) (Goodfellow *et al.*, 2014) and more efficient sampling mechanisms than autoregressive models (Larochelle and Murray, 2011). Much of the recent literature has focused on applying VAEs on image generation tasks (Lucic *et al.*, 2018; Ham *et al.*, 2018; Huang *et al.*, 2018) and devising new encoder–decoder architectures (Van den Oord *et al.*, 2017; Jin *et al.*, 2018).

Despite this attention from the community, the VAE framework is still far from delivering the promised generative mechanism in many real-world scenarios. In fact, VAEs tend to generate blurry samples, a condition which has been attributed to using overly simplistic distributions for the prior (Tomczak and Welling, 2018); restrictiveness of the Gaussian assumption for the stochastic architecture (Dai and Wipf, 2019); or over-regularization induced by the KL divergence term in the VAE objective (Tolstikhin *et al.*, 2017) (see Figure 6.1). Moreover, the VAE objective itself poses several challenges as it admits trivial solutions that decouple the latent space from the input (Chen *et al.*, 2017b; Zhao *et al.*, 2017b), leading to the posterior collapse phenomenon in conjunction with powerful decoders (Van den Oord *et al.*, 2017). Training a VAE requires approximating expectations by sampling at the cost of increased variance in gradients (Tucker *et al.*, 2017; Burda *et al.*, 2015), making initialization, validation and annealing of hyperparameters fundamental in practice (Bowman *et al.*, 2016; Bauer and Mnih, 2019; Higgins *et al.*, 2017). Lastly, even after a satisfactory convergence of the objective, the learned aggregated posterior distribution rarely matches the assumed latent prior in practice (Kingma *et al.*, 2016; Bauer and Mnih, 2019; Dai and Wipf, 2019), ultimately hurting the quality of generated samples.

In this work, we tackle these shortcomings by reformulating the VAE into a generative modeling scheme that scales better, is simpler to optimize, and most importantly, produces higher-quality samples. We do so based on the ovservation that under common distributional assumptions made for VAEs, training a stochastic encoder–decoder pair does not differ in practice from training a deterministic architecture where noise is added to the decoder's input to enforce a smooth latent space. We investigate how to substitute this noise injection mechanism with other regularization schemes in our deterministic *Regularized Autoencoders* (RAEs), and we analyze how we can learn a meaningful latent space without forcing it to conform to a given prior distribution. We equip RAEs with a generative mechanism through a simple and efficient ex-post density estimation step on the learned latent space which leads to improved image quality that surpasses VAEs and stronger alternatives such as Wasserstein Autoencoders (WAEs) (Tolstikhin *et al.*, 2017). In summary, our contributions are as follows:

1. we introduce the RAE framework for generative modeling,
2. we propose an ex-post density estimation scheme that greatly improves sample

quality for the VAE, WAE and RAE without the need for additional training,

3. we conduct a rigorous empirical evaluation on several common image datasets (MNIST, CIFAR-10, CelebA), assaying reconstruction, random samples and interpolation quality for VAE, WAE and RAE,

4. we achieve state-of-the-art FID scores for the above datasets in a non-adversarial setting.

The chapter is organized as follows. In Section 6.2 we introduce the VAE framework and discuss assumptions, practical implementations and limitations, leading to the introduction of our simplified deterministic and regularized framework (Section 6.3), interpreting explicit regularization as constrained optimization under certain parametric assumptions (Section 6.4). After discussing ex-post density estimation and related works in Sections 6.5 and 6.6, we present experiments in Section 6.7 before we close with our final conclusions.

6.2 Variational Autoencoders

For a general discussion, we consider a collection of high-dimensional *i.i.d.* samples $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ drawn from the true data distribution $p_{\mathsf{data}}(\mathbf{x})$ over a random variable $\mathbf{X}$ taking values in the input space. The aim of generative modeling is to learn from $\mathcal{X}$ a mechanism to draw new samples $\mathbf{x}_{\mathsf{new}} \sim p_{\mathsf{data}}$.

Variational Autoencoders provide a powerful latent variable framework to infer such a mechanism. The generative process of the VAE is defined as

$$\mathbf{z}_{\mathsf{new}} \sim p(\mathbf{Z}), \qquad \mathbf{x}_{\mathsf{new}} \sim p_\theta(\mathbf{X} \,|\, \mathbf{Z} = \mathbf{z}_{\mathsf{new}}) \tag{6.1}$$

where $p(\mathbf{Z})$ is a fixed prior distribution over a low-dimensional latent variable $\mathbf{Z}$. A stochastic decoder

$$D_\theta(\mathbf{z}) = \mathbf{x} \sim p_\theta(\mathbf{x} \,|\, \mathbf{z}) = p(\mathbf{X} \,|\, g_\theta(\mathbf{z})) \tag{6.2}$$

links the latent space to the input space through the *likelihood*, where g_θ is an expressive non-linear function parameterized by θ.[1] As a result, a VAE estimates $p_{\mathsf{data}}(\mathbf{x})$ as the infinite mixture model $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x} \,|\, \mathbf{z}) p(\mathbf{z}) d\mathbf{z}$. At the same time, the input space is mapped to the latent space via a stochastic encoder

$$D_\theta(\mathbf{x}) = \mathbf{z} \sim q_\phi(\mathbf{z} \,|\, \mathbf{x}) = q(\mathbf{Z} \,|\, f_\phi(\mathbf{x})) \tag{6.3}$$

where $q_\phi(\mathbf{z} \,|\, \mathbf{x})$ is the *posterior* distribution given by a second function f_ϕ parameterized by ϕ.

[1] With slight abuse of notation, we use lowercase letters for both random variables and their realizations, *e.g.* $p_\theta(\mathbf{x} \,|\, \mathbf{z})$ instead of $p(\mathbf{X} \,|\, \mathbf{Z} = \mathbf{z})$, when it is clear to discriminate between the two.

Computing the marginal log-likelihood $\log p_\theta(\mathbf{x})$ is generally intractable. We therefore follow a variational approach, maximizing the evidence lower bound (ELBO) for a sample $\mathbf{x}$:

$$\begin{aligned}&\log p_\theta(\mathbf{x}) \geq \mathsf{ELBO}(\phi,\theta,\mathbf{x}) = \\ &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - \mathbb{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))\end{aligned} \tag{6.4}$$

Maximizing Eq. 6.4 over data $\mathcal{X}$ w.r.t. model parameters ϕ, θ corresponds to minimizing the loss

$$\arg\min_{\phi,\theta} \mathbb{E}_{\mathbf{x}\sim p_{\mathsf{data}}} \mathcal{L}_{\mathsf{ELBO}} = \mathbb{E}_{\mathbf{x}\sim p_{\mathsf{data}}} \mathcal{L}_{\mathsf{REC}} + \mathcal{L}_{\mathsf{KL}} \tag{6.5}$$

where $\mathcal{L}_{\mathsf{REC}}$ and $\mathcal{L}_{\mathsf{KL}}$ are defined for a sample $\mathbf{x}$ as follows:

$$\mathcal{L}_{\mathsf{REC}} = -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) \tag{6.6}$$

$$\mathcal{L}_{\mathsf{KL}} = \mathbb{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \tag{6.7}$$

Intuitively, the reconstruction loss $\mathcal{L}_{\mathsf{REC}}$ takes into account the quality of autoencoded samples $\mathbf{x}$ through $D_\theta(D_\theta(\mathbf{x}))$, while the KL-divergence term $\mathcal{L}_{\mathsf{KL}}$ encourages $q_\phi(\mathbf{z}|\mathbf{x})$ to match the prior $p(\mathbf{z})$ for each $\mathbf{z}$ which acts as a regularizer during training (Hoffman and Johnson, 2016). For the purpose of generating high-quality samples, a balance between these two loss terms must be found during training, see Figure 6.1.

6.2.1 Practice and shortcomings of VAEs

To fit a VAE to data through Eq. 6.5 one has specify the parametric forms for $p(\mathbf{z})$, $q_\phi(\mathbf{z}|\mathbf{x})$, $p_\theta(\mathbf{x}|\mathbf{z})$, and hence the deterministic mappings f_ϕ and g_θ. In practice, the choice for the above distributions is guided by trading off computational complexity with model expressiveness.

In the most common formulation of the VAE, $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z})$ are assumed to be Gaussian

$$D_\theta(\mathbf{x}) \sim \mathcal{N}(\mathbf{Z}|\mu_\phi(\mathbf{x}), \mathsf{diag}(\sigma_\phi(\mathbf{x}))) \tag{6.8}$$

$$D_\theta(D_\theta(\mathbf{x})) \sim \mathcal{N}(\mathbf{X}|\mu_\theta(\mathbf{z}), \mathsf{diag}(\sigma_\theta(\mathbf{z}))) \tag{6.9}$$

with means μ_ϕ, μ_θ and covariance parameters $\sigma_\phi, \sigma_\theta$ given by f_ϕ and g_θ. In practical implementations, the covariance of the decoder is set to the identity matrix for all $\mathbf{z}$, *i.e.* $\sigma_\theta(\mathbf{z}) = 1$ (Dai and Wipf, 2019). The expectation of $\mathcal{L}_{\mathsf{REC}}$ in Eq. 6.6 is then approximated via k Monte Carlo point estimates. We find clear evidence that larger values lead to improvements in training as shown in Figure 6.2. Nevertheless, only a one-sample approximation is carried out in practice (Kingma and Welling, 2014) since requirements to memory and computation scale linearly with k. With this approximation, the computation of $\mathcal{L}_{\mathsf{REC}}$ is given by the mean squared error between input samples and their mean

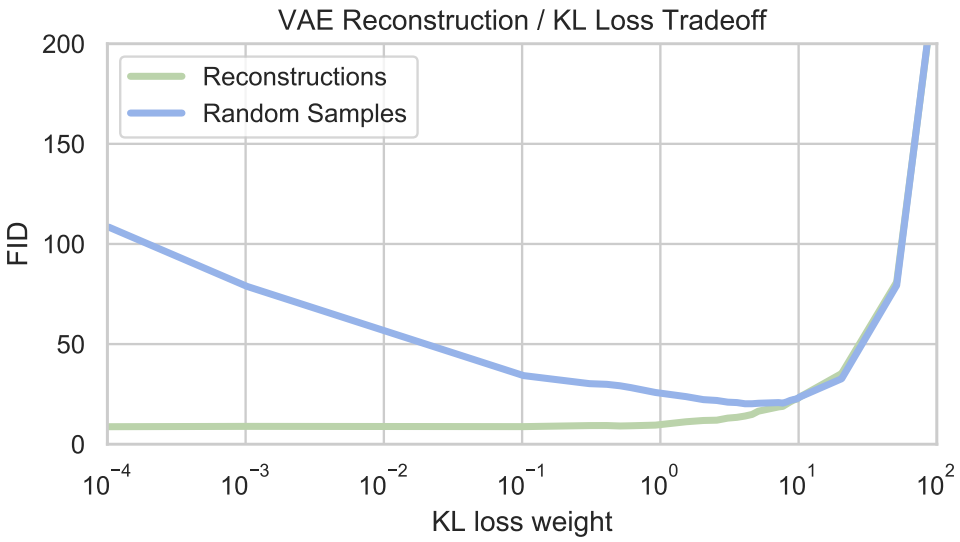


Figure 6.1: Reconstruction and random sample quality (FID, y-axis, lower is better) of a VAE on MNIST for different tradeoffs between $\mathcal{L}_{\text{REC}}$ and $\mathcal{L}_{\text{KL}}$ (x-axis, see Eq. 6.5). Higher weights for $\mathcal{L}_{\text{KL}}$ improve random samples but hurt reconstruction. Enforcing structure in the VAE latent space leads to a penalty in quality.

reconstructions μ_θ through the deterministic decoder:

$$\mathcal{L}_{\text{REC}} = ||\mathbf{x} - \mu_\theta(E_\phi(\mathbf{x}))||_2^2 \tag{6.10}$$

Gradients w.r.t. the encoder parameters ϕ are computed through the expectation of $\mathcal{L}_{\text{REC}}$ in Eq. 6.6 via the reparametrization trick (Kingma and Welling, 2014) where the stochasticity of D_θ is relegated to an auxiliary random variable ε which does not depend on ϕ:

$$D_\theta(\mathbf{x}) = \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x}) \odot \varepsilon \qquad \varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{6.11}$$

where $\odot$ denotes the Hadamard product. An additional simplifying assumption involves fixing the prior $p(\mathbf{z})$ to be a D-dimensional isotropic Gaussian $\mathcal{N}(\mathbf{Z}\,|\,\mathbf{0}, \mathbf{I})$. For this choice, the KL-divergence for a sample $\mathbf{x}$ is given in closed form:

$$2\mathcal{L}_{\text{KL}} = ||\mu_\phi(\mathbf{x})||_2^2 + D + \sum_i^D \sigma_\phi(\mathbf{x})_i - \log \sigma_\phi(\mathbf{x})_i \tag{6.12}$$

While the chosen assumptions make VAEs easy to implement, the stochasticity in the encoder and decoder has been deemed to be responsible for the "blurriness" in VAE samples (Makhzani *et al.*, 2016; Tolstikhin *et al.*, 2017; Dai and Wipf, 2019). Furthermore, the optimization problem as shown in Eq. 6.5 presents some further challenges. Imposing

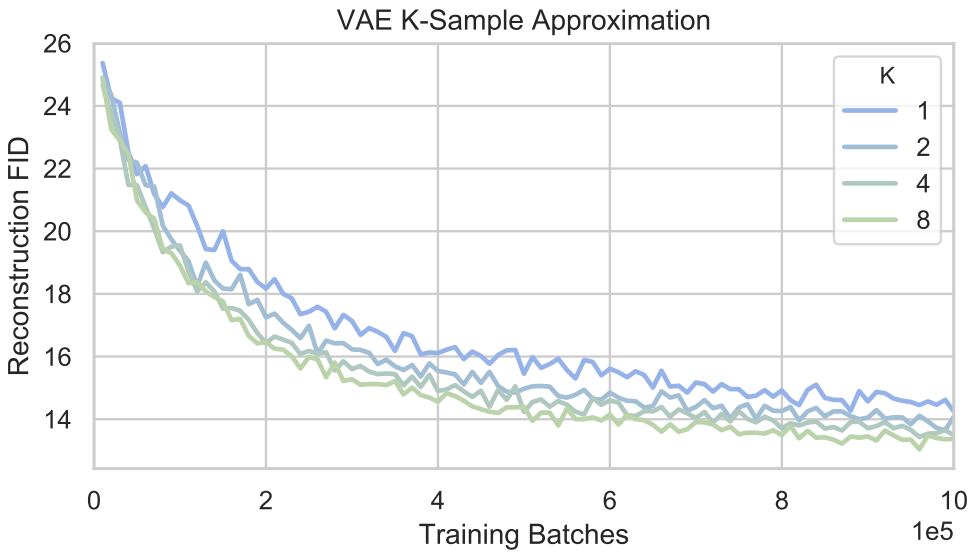


Figure 6.2: Test reconstruction quality for a VAE trained on MNIST with different numbers of samples in the latent space as in Eq. 6.8 measured by FID (lower is better). Larger numbers of samples clearly improve training, however, the increased accuracy comes with larger requirements for memory and computation. In practice, the most common choice is therefore $k = 1$.

a strong weight on the $\mathcal{L}_{\text{KL}}$ term during optimization can dominate $\mathcal{L}_{\text{ELBO}}$, having the effect of *over-regularization* which leads to blurred samples, see Figure 6.1. Heuristics to avoid this include gradually annealing the importance of $\mathcal{L}_{\text{KL}}$ during training (Bowman *et al.*, 2016; Bauer and Mnih, 2019) and manually fine-tuning the balance between the losses.

Even after employing the full array of approximations and "tricks" to reach convergence of Eq. 6.5 for a satisfactory set of parameters, there is no guarantee that the learned latent space is distributed according to the assumed prior distribution. In other words, the aggregated posterior distribution $q_\phi(\mathbf{z}) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}} q(\mathbf{z}|\mathbf{x})$ has been shown not to conform well to $p(\mathbf{z})$ after training (Tolstikhin *et al.*, 2017; Bauer and Mnih, 2019; Dai and Wipf, 2019). This critical issue severely hinders the generative mechanism of a VAE (Eq. 6.1) since latent codes sampled from $p(\mathbf{z})$ (instead of $q(\mathbf{z})$) might lead to regions of the latent space that are previously unseen to D_θ during training. The result is blurry out-of-distribution samples. We analyze solutions to this problem in Section 6.5.

6.2.2 Constant-Variance Encoders

Analogously to what is generally done for decoders, we also investigate fixing the variance of $q_\phi(\mathbf{z}|\mathbf{x})$ to be constant for all $\mathbf{x}$. This simplifies the computation of D_θ from Eq. 6.11

to

$$D_\theta^{\mathsf{CV}}(\mathbf{x}) = \mu_\phi(\mathbf{x}) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I}) \tag{6.13}$$

where σ is a fixed scalar. At the same time, the KL loss term in Eq. 6.12 simplifies (up to constants) to

$$2\mathcal{L}_{\mathsf{KL}}^{\mathsf{CV}} = ||\mu_\phi(\mathbf{x})||_2^2 \tag{6.14}$$

Constant-Variance VAEs (CV-VAEs) have been previously applied in applications of adversarial robustness (Ghosh *et al.*, 2019) and variational image compression (Ballé *et al.*, 2017) but to the best of our knowledge, there is no systematic study of CV-VAEs in the literature. As noted in (Ghosh *et al.*, 2019), treating σ_ϕ as a constant impairs the assumption of $p(\mathbf{z})$ to be an isotropic Gaussian which demands a more complex prior structure over $\mathbf{Z}$. We address this mismatch in Section 6.5.

6.3 Deterministic Regularized Autoencoders

As described in Section 6.2, autoencoding in VAEs is defined in a probabilistic fashion: D_θ and D_θ map data points not to a single point, but rather to parameterized distributions as shown in Equations 6.8 and 6.9. However, the practical implementation of the VAE admits a deterministic view for this probabilistic mechanism.

A glance at the autoencoding mechanism of the VAE is revealing. The encoder maps a data point $\mathbf{x}$ to a mean $\mu_\phi(\mathbf{x})$ and variance $\sigma_\phi(\mathbf{x})$ in the latent space via the reparametrization trick given in Eq. 6.11. The input to the decoder is then simply the mean $\mu_\phi(\mathbf{x})$ augmented with random Gaussian noise scaled by $\sigma_\phi(\mathbf{x})$. In the CV-VAE, this relationship is even more obvious, as the magnitude of the noise is fixed for all data points (Eq. 6.13). In this light, a VAE can be seen as a *deterministic* autoencoder where Gaussian noise is added to the decoder's input.

Using random noise injection to regularize neural networks during training is a well-known technique that dates back several decades (Sietsma and Dow, 1991; An, 1996). The addition of noise implicitly smooths the function learned by the network. Since this procedure also adds noise to the gradients, we propose to substitute noise injection with an explicit regularization scheme for the decoder network. Note that from a generative perspective, this is motivated by the goal to learn a smooth latent space where similar data points $\mathbf{x}$ are mapped to similar latent codes $\mathbf{z}$, and small variations in $\mathbf{Z}$ lead to reconstructions by D_θ that vary only slightly.

By removing the noise injection mechanism from the CV-VAE, we are effectively left with a deterministic *Regularized Autoencoder* (RAE) that can be coupled with any type of explicit regularization for the decoder to enforce a smooth latent space. Training a RAE thus involves minimizing the simplified loss

$$\mathcal{L}_{\mathsf{RAE}} = \mathcal{L}_{\mathsf{REC}} + \beta\mathcal{L}_{\mathsf{KL}}^{\mathsf{RAE}} + \lambda\mathcal{L}_{\mathsf{REG}} \tag{6.15}$$

where $\mathcal{L}_{\mathsf{REG}}$ represents the explicit regularizer for D_θ (see Section 6.3.1) and $\mathcal{L}_{\mathsf{KL}}^{\mathsf{RAE}} = {}^1\!/_2||\mathbf{z}||_2^2$ from Eq. 6.14 is equivalent to constraining the size of the learned space. Note that for RAEs, no sampling approximation of $\mathcal{L}_{\mathsf{REC}}$ is required, thus relieving the need for more samples from $q_\phi(\mathbf{z}|\mathbf{x})$ to achieve better image quality (see Figure 6.2).

6.3.1 Regularization Schemes for RAEs

Among possible choices for a mechanism to use for $\mathcal{L}_{\mathsf{REG}}$, a first obvious candidate is Tikhonov regularization (Tikhonov and Arsenin, 1977) since is known to be related to the addition of low-magnitude input noise (Bishop, 2006). Training a RAE within this framework thus amounts to adopting

$$\mathcal{L}_{\mathsf{REG}} = \mathcal{L}_{\mathsf{L}_2} = ||\theta||_2^2 \tag{6.16}$$

where $\mathcal{L}_{\mathsf{L}_2}$ effectively applies weight decay on the decoder parameters θ.

Another avenue comes from the recent GAN literature where regularization is a hot topic (Kurach *et al.*, 2019) and where injecting noise to the input of the adversarial discriminator has led to improved performance in a technique called *instance noise* (Sønderby *et al.*, 2017). To enforce Lipschitz continuity on adversarial discriminators, weight clipping has been proposed (Arjovsky *et al.*, 2017), which is however known to significantly slow down training. More successfully, a *gradient penalty* on the discriminator can be used similar to (Gulrajani *et al.*, 2017; Mescheder *et al.*, 2018), yielding the objective

$$\mathcal{L}_{\mathsf{REG}} = \mathcal{L}_{\mathsf{GP}} = ||\nabla D_\theta(D_\theta(\mathbf{x}))||_2^2 \tag{6.17}$$

which encourages small L_2 norm of the gradient of the decoder w.r.t. its input.

Additionally, spectral normalization (SN) has been proposed as an alternative way to bound the Lipschitz norm of an adversarial discriminator, showing very promising results for GAN training (Miyato *et al.*, 2018). SN normalizes the weight matrix θ_ℓ for each layer in the decoder by dividing it by an estimate of its largest singular value:

$$\theta_\ell^{\mathsf{SN}} = \theta_\ell / \mathsf{s}(\theta_\ell) \tag{6.18}$$

where $\mathsf{s}(\theta_\ell)$ is the current estimate obtained through the power method.

Lastly, in light of recent success stories of deep neural networks *without* explicit regularization achieving state-of-the-art results (Zhang *et al.*, 2017a; Zagoruyko and Komodakis, 2016), it is intriguing to question the need to explicitly regularize the decoder in order to obtain a meaningful latent space. The assumption here is that techniques such as dropout (Srivastava *et al.*, 2014), batch normalization (Ioffe and Szegedy, 2015), adding noise during training (An, 1996), or early stopping in conjunction with novel architectural developments implicitly regularize the networks enough to smoothen the latent space. Therefore, as a natural baseline to the $\mathcal{L}_{\mathsf{RAE}}$ objectives introduced above, we also consider

the RAE framework without $\mathcal{L}_{\mathsf{REG}}$ and $\mathcal{L}_{\mathsf{KL}}^{\mathsf{RAE}}$, *i.e.* a standard deterministic autoencoder optimizing $\mathcal{L}_{\mathsf{REC}}$ only.

6.4 A Probabilistic Derivation of Smoothing

In this section, we propose an alternative view on enforcing smoothness on the output of D_θ by augmenting the ELBO optimization problem for VAEs with an explicit constraint. While we keep the Gaussianity assumptions over a stochastic D_θ and $p(\mathbf{z})$ for convenience, we are not fixing a parametric form for $q_\phi(\mathbf{z}|\mathbf{x})$ yet. We will then discuss how some parametric restrictions over $q_\phi(\mathbf{z}|\mathbf{x})$ indeed lead to a variation of the RAE framework in Eq. 6.15, specifically as the introduction of $\mathcal{L}_{\mathsf{GP}}$ as a regularizer of a deterministic version of the CV-VAE.

To start, we can augment the minimization in Eq. 6.5 as:

$$\begin{aligned} \arg\min_{\phi,\theta} \; & \mathbb{E}_{\mathbf{x}\sim p_{\mathsf{data}}(\mathbf{X})}\, \mathcal{L}_{\mathsf{REC}} + \mathcal{L}_{\mathsf{KL}} \\ \text{s.t. } & ||D_\theta(\mathbf{z}_1) - D_\theta(\mathbf{z}_2)||_p < \varepsilon \\ \forall\, \mathbf{z}_1, \mathbf{z}_2 \sim q_\phi(\mathbf{z}|\mathbf{x}) \quad & \forall \mathbf{x} \sim p_{\mathsf{data}}(\mathbf{X}) \end{aligned} \tag{6.19}$$

where $D_\theta(\mathbf{z}) = \mu_\theta(E_\phi(\mathbf{x}))$ and the constraint on the decoder encodes that the output has to vary, in the sense of an L_p norm, only by a small amount ε for any two possible draws from the encoder. Using the mean value theorem, there exists a $\tilde{\mathbf{z}} \sim q_\phi(\mathbf{z}|\mathbf{x})$ such that the left term in the constraint can be bounded as:

$$\begin{aligned} ||D_\theta(\mathbf{z}_1) - D_\theta(\mathbf{z}_2)||_p &= \nabla D_\theta(\tilde{\mathbf{z}}) \cdot ||\mathbf{z}_1 - \mathbf{z}_2||_p \\ &\leq sup\{||\nabla D_\theta(\mathbf{z})||_p\} \cdot sup\{||\mathbf{z}_1 - \mathbf{z}_2||_p\} \end{aligned} \tag{6.20}$$

where we take the supremum of possible gradients of D_θ as well as the supremum of a measure of the support of $q_\phi(\mathbf{z}|\mathbf{x})$. From this form of the smoothness constraint, it is apparent why the choice of a parametric form for $q_\phi(\mathbf{z}|\mathbf{x})$ can be impactful during training. For a compactly supported isotropic PDF $q_\phi(\mathbf{z}|\mathbf{x})$, the extension of the support $sup\{||\mathbf{z}_1 - \mathbf{z}_2||_p\}$ would be dependent on $\mathbb{H}(q_\phi(\mathbf{z}|\mathbf{x}))$, the entropy of $q_\phi(\mathbf{z}|\mathbf{x})$, through some functional r. For instance, a uniform posterior over a hypersphere in $\mathbf{z}$ would ascertain $r(\mathbb{H}(q_\phi(\mathbf{z}|\mathbf{x}))) \cong e^{\mathbb{H}(q_\phi(\mathbf{z}|\mathbf{x}))/n}$ where n is the dimensionality of the latent space.

Intuitively, one would look for parametric distributions that do not favor overfitting, *e.g.* degenerating in Dirac-deltas (minimal entropy and support) along any dimensions. To this end, an isotropic nature of $q_\phi(\mathbf{z}|\mathbf{x})$ would favor such a robustness against decoder over-fitting. We can now rewrite the constraint as

$$r(\mathbb{H}(q_\phi(\mathbf{z}|\mathbf{x}))) \cdot sup\{||\nabla D_\theta(\mathbf{z})||_p\} < \varepsilon \tag{6.21}$$

The $\mathcal{L}_{\mathsf{KL}}$ term can be expressed in terms of $\mathbb{H}(q_\phi(\mathbf{z}|\mathbf{x}))$, by decomposing it as $\mathcal{L}_{\mathsf{KL}} =$

$\mathcal{L}_{\mathsf{CE}} - \mathcal{L}_{\mathsf{H}}$, where $\mathcal{L}_{\mathsf{H}} = \mathbb{H}(q_\phi(\mathbf{z}|\mathbf{x}))$ and $\mathcal{L}_{\mathsf{CE}} = \mathbb{H}(q_\phi(\mathbf{z}|\mathbf{x}), p(\mathbf{z}))$ represents a cross-entropy term. Therefore, the constrained problem in Eq. 6.19 can be written in a Lagrangian formulation by including Eq. 6.21:

$$\arg\min_{\phi,\theta} \mathbb{E}_{\mathbf{x}\sim p_{\mathsf{data}}} \mathcal{L}_{\mathsf{REC}} + \mathcal{L}_{\mathsf{CE}} - \mathcal{L}_{\mathsf{H}} + \lambda \mathcal{L}_{\mathsf{LANG}} \tag{6.22}$$

where $\mathcal{L}_{\mathsf{LANG}} = r(\mathbb{H}(q_\phi(\mathbf{z}|\mathbf{x}))) * ||\nabla D_\theta(\mathbf{z})||_p$. We argue that a reasonable simplifying assumption for $q_\phi(\mathbf{z}|\mathbf{x})$ is to fix $\mathbb{H}(q_\phi(\mathbf{z}|\mathbf{x}))$ to a single constant for all samples $\mathbf{x}$. Intuitively, this can be understood as fixing the variance in $q_\phi(\mathbf{z}|\mathbf{x})$ as we did for the CV-VAE in Section 6.2.2. With this simplification, Eq. 6.22 further reduces to

$$\arg\min_{\phi,\theta} \mathbb{E}_{\mathbf{x}\sim p_{\mathsf{data}}(\mathbf{X})} \mathcal{L}_{\mathsf{REC}} + \mathcal{L}_{\mathsf{CE}} + \lambda ||\nabla D_\theta(\mathbf{z})||_p \tag{6.23}$$

We can see that $||\nabla D_\theta(\mathbf{z})||_p$ results to be the gradient penalty $\mathcal{L}_{\mathsf{GP}}$ and $\mathcal{L}_{\mathsf{CE}} = ||\mathbf{z}||_2^2$ corresponds to $\mathcal{L}_{\mathsf{KL}}^{\mathsf{RAE}}$, thus recovering our RAE framework as presented in Eq. 6.15.

6.5 Ex-Post Density Estimation

By removing stochasticity and ultimately, the KL divergence term $\mathcal{L}_{\mathsf{KL}}$ from RAEs, we have simplified the original VAE objective at the cost of detaching the encoder from the prior $p(\mathbf{z})$ over the latent space. This implies i) we cannot ensure that the latent space $\mathbf{Z}$ is distributed according to a simple distribution anymore (*e.g.* isotropic Gaussian) and consequently, ii) we lose the simple mechanism provided by $p(\mathbf{z})$ to sample from $\mathbf{Z}$ as in Eq. 6.1.

As discussed in Section 6.2.1, issue i) is compromising the VAE framework in any case in practice as reported in several works (Dai and Wipf, 2019; Rosca *et al.*, 2018; Hoffman and Johnson, 2016). To fix this, some works extend the VAE objective by encouraging the aggregated posterior to match $p(\mathbf{z})$ (Tolstikhin *et al.*, 2017) or by utilizing more complex priors (Kingma *et al.*, 2016; Bauer and Mnih, 2019; Tomczak and Welling, 2018).

To overcome both i) and ii), we instead propose to employ *ex-post density estimation* over $\mathbf{Z}$. We fit a density estimator denoted as $q_\delta(\mathbf{z})$ to $\{\mathbf{z} = D_\theta(\mathbf{x}) | \mathbf{x} \in \mathcal{X}\}$. This simple approach not only fits our RAE framework well, but it can also be readily adopted for any VAE or variants thereof such as the WAE as a practical remedy to the aggregated posterior mismatch without adding any computational overhead to the costly training phase.

The choice of $q_\delta(\mathbf{z})$ needs to trade-off *expressiveness* – to provide a good fit of an arbitrary space for $\mathbf{Z}$ – with *simplicity* – to improve generalization. Indeed, placing a Dirac distribution on each latent point $\mathbf{z}$ would allow the decoder to output only training sample reconstructions. Striving for simplicity and in order to show the effectiveness of the proposed ex-post density estimation scheme, we compare a full covariance multivariate Gaussian with a 10-component Gaussian mixture model (GMM) in our experiments.

6.6 Related works

Many works have focused on diagnosing the VAE framework, the terms in its objective (Alemi *et al.*, 2018; Hoffman and Johnson, 2016; Zhao *et al.*, 2017b), and ultimately augmenting it to solve optimization issues (Rezende and Viola, 2018; Dai and Wipf, 2019). With RAE, we argue that a simpler deterministic framework can be competitive for generative modeling.

Deterministic denoising (Vincent *et al.*, 2008) and contractive autoencoders (CAEs) (Rifai *et al.*, 2011) have received attention in the past for their ability to capture a smooth data manifold. Heuristic attempts to equip them with a generative mechanism include MCMC schemes (Rifai *et al.*, 2012; Bengio *et al.*, 2013). However, they are hard to diagnose for convergence, require a considerable effort in tuning (Cowles and Carlin, 1996), and have not scaled beyond MNIST, leading to them being superseded by VAEs. While in spirit the proposed RAE is similar, $\mathcal{L}_{\mathsf{GP}}$ requires much less computational effort than computing the Jacobian for CAEs (Rifai *et al.*, 2011).

Approaches to cope with the aggregated posterior mismatch involve fixing a more expressive form for $p(\mathbf{z})$ (Kingma *et al.*, 2016; Bauer and Mnih, 2019) therefore altering the VAE objective and requiring considerable additional computational efforts. Estimating the latent space of a VAE with a second VAE (Dai and Wipf, 2019) reintroduces many of the optimization shortcomings discussed for VAEs and is much more expensive in practice compared to fitting a simple $q_\delta(\mathbf{z})$ after training.

GANs (Goodfellow *et al.*, 2014) have received widespread attention for their ability to produce sharp samples. Despite theoretical and practical advances (Arjovsky *et al.*, 2017), the training procedure of GANs is still unstable, sensitive to hyperparameters, and prone to the *mode collapse* problem (Lucic *et al.*, 2018; Sajjadi *et al.*, 2018a,c).

Adversarial Autoencoders (AAE) (Makhzani *et al.*, 2016) add a discriminator to a deterministic encoder–decoder pair, leading to sharper samples at the expense of higher computational overhead and the introduction of instabilities caused by the adversarial nature of the training process. Wasserstein Autoencoders (WAE) (Tolstikhin *et al.*, 2017) have been introduced as a generalization of AAEs by casting autoencoding as an optimal transport (OT) problem. Both stochastic and deterministic models can be trained by minimizing a relaxed OT cost function employing either an adversarial loss term or the maximum mean discrepancy score between $p(\mathbf{z})$ and $q_\phi(\mathbf{z})$ as a regularizer in place of $\mathcal{L}_{\mathsf{KL}}$.

Within the RAE framework, we look at this problem from a different perspective: instead of explicitly imposing a simple structure on $\mathbf{Z}$ that might impair the ability to fit high-dimensional data during training, we propose to model the latent space by an ex-post density estimation step.

	MNIST	CIFAR-10	CELEBA
Encoder	$x \in \mathcal{R}^{32\times32}$	$x \in \mathcal{R}^{32\times32}$	$x \in \mathcal{R}^{64\times64}$
	$\text{Conv}_{128} \to \text{BN} \to \text{ReLU}$	$\text{Conv}_{128} \to \text{BN} \to \text{ReLU}$	$\text{Conv}_{128} \to \text{BN} \to \text{ReLU}$
	$\text{Conv}_{256} \to \text{BN} \to \text{ReLU}$	$\text{Conv}_{256} \to \text{BN} \to \text{ReLU}$	$\text{Conv}_{256} \to \text{BN} \to \text{ReLU}$
	$\text{Conv}_{512} \to \text{BN} \to \text{ReLU}$	$\text{Conv}_{512} \to \text{BN} \to \text{ReLU}$	$\text{Conv}_{512} \to \text{BN} \to \text{ReLU}$
	$\text{Conv}_{1024} \to \text{BN} \to \text{ReLU}$	$\text{Conv}_{1024} \to \text{BN} \to \text{ReLU}$	$\text{Conv}_{1024} \to \text{BN} \to \text{ReLU}$
	$\text{Flatten} \to \text{FC}_{16\times M}$	$\text{Flatten} \to \text{FC}_{128\times M}$	$\text{Flatten} \to \text{FC}_{64\times M}$
Decoder	$z \in \mathcal{R}^{16} \to \text{FC}_{8\times8\times1024}$	$z \in \mathcal{R}^{128} \to \text{FC}_{8\times8\times1024}$	$z \in \mathcal{R}^{64} \to \text{FC}_{8\times8\times1024}$
	$\text{BN} \to \text{ReLU}$	$\text{BN} \to \text{ReLU}$	$\text{BN} \to \text{ReLU}$
	$\text{ConvT}_{512} \to \text{BN} \to \text{ReLU}$	$\text{ConvT}_{512} \to \text{BN} \to \text{ReLU}$	$\text{ConvT}_{512} \to \text{BN} \to \text{ReLU}$
	$\text{ConvT}_{256} \to \text{BN} \to \text{ReLU}$	$\text{ConvT}_{256} \to \text{BN} \to \text{ReLU}$	$\text{ConvT}_{256} \to \text{BN} \to \text{ReLU}$
	ConvT_{1}	ConvT_{1}	$\text{ConvT}_{128} \to \text{BN} \to \text{ReLU}$
			ConvT_{1}

Table 6.1: RAE model architecture. Conv_n represents a convolutional layer with n filters. All convolutions Conv_n and transposed convolutions ConvT_n have a filter size of 4×4 for MNIST and CIFAR-10 and 5×5 for CELEBA. They all have a stride of size 2 except for the last convolutional layer in the decoder. Finally, $M = 1$ for all models except for the VAE which has $M = 2$ as the encoder has to produce both mean and variance for each input.

6.7 Experiments

In this Section, we investigate the performance of several VAE and RAE variants on MNIST (LeCun *et al.*, 1998), CIFAR-10 (Krizhevsky and Hinton, 2009) and CelebA (Liu *et al.*, 2015a). We measure three qualities for each model: held-out sample reconstruction quality, random sample quality, and interpolation quality. While reconstructions give us a lower bound on the best quality achievable by the generative model, random sample quality indicates how well the model generalizes. Finally, interpolation quality sheds light on the structure of the learned latent space.

6.7.1 Models

We compare the the proposed RAE model with the gradient penalty (RAE-GP), with weight decay (RAE-L2), and with spectral normalization (RAE-SN). Additionally, we consider two models for which we either add only the latent code regularizer $\mathcal{L}_{\mathsf{KL}}^{\mathsf{RAE}}$ to $\mathcal{L}_{\mathsf{REC}}$ (RAE), or no explicit regularization at all (AE). As baselines, we further employ a regular VAE, the constant-variance VAE (CV-VAE) for comparison, and finally, a Wasserstein Autoencoder (WAE) with the MMD loss as a state-of-the-art alternative.

Aiming for a fair comparison, we employ the same network architecture for all models. We largely follow the models adopted in (Tolstikhin *et al.*, 2017) with the difference that we consistently apply batch normalization (Ioffe and Szegedy, 2015) for all models as

	MNIST				CIFAR				CelebA			
	Rec.	Samples			Rec.	Samples			Rec.	Samples		
		$\mathcal{N}$	GM	Int.		$\mathcal{N}$	GM	Int.		$\mathcal{N}$	GM	Int.
VAE	18.3	**19.2**	17.7	18.2	58.0	106.3	103.8	88.6	39.1	48.1	45.5	44.5
CV-VAE	15.2	33.8	17.9	25.1	37.7	94.8	86.6	69.7	40.4	48.9	49.3	45.0
WAE	10.0	20.4	9.4	**14.3**	36.0	117.4	93.5	76.9	34.8	53.7	42.7	41.0
RAE-GP	14.0	22.2	11.5	15.3	32.2	83.1	76.3	64.1	39.7	116.3	45.6	47.0
RAE-L2	10.5	22.2	**8.7**	14.5	32.2	**80.8**	**74.2**	62.5	43.5	51.1	48.0	46.0
RAE-SN	15.7	19.7	11.7	15.2	27.6	84.3	75.3	63.6	36.0	**44.7**	**41.0**	**39.5**
RAE	11.7	23.9	9.8	14.7	29.1	83.9	76.3	63.3	40.2	48.2	44.7	43.7
AE	13.0	58.7	10.7	17.1	30.5	84.7	76.5	**61.6**	40.8	127.9	45.1	51.0

Table 6.2: Evaluation of all models by FID (lower numbers are better, best model in bold). We evaluate each model by Rec.: test sample reconstruction; $\mathcal{N}$: random samples generated according to the prior distribution $p(\mathbf{z})$ (for VAE / WAE) or by fitting a Gaussian to $q_\delta(\mathbf{z})$ (for the remaining models); GM: random samples generated by fitting a mixture of 10 Gaussians in the latent space; Int.: mid-point interpolation between random pairs of test reconstructions. Note that our (less constrained) RAE models are competitive with or outperform the VAE and WAE throughout the evaluation. Surprisingly, interpolations do not suffer from the lack of explicit prior on the latent space in our models. Furthermore, the unregularized AE achieves very good FID scores when combined with the proposed ex-post density estimation technique.

we found it to improve performance across the range. The latent space dimension is 16 for MNIST, 128 for CIFAR-10 and 64 for CelebA. Further details about the network architecture and training procedure are given in Section 6.8.

6.8 Network architecture and Training Details

For all experiments, we use the Adam optimizer with a starting learning rate of 10^{-3} which is cut in half every time the validation loss plateaus. All models are trained for a maximum of 100 epochs on MNIST and CIFAR and 70 epochs on CelebA. We use mini-batch size of 100 and pad MNIST digits with zeros to make the size 32×32.

We use official train, validation and test splits of CelebA. For MNIST and CIFAR, we set aside 10k train samples for validation. For random sample evaluation, we draw samples from $\mathcal{N}(0,I)$ for VAE and WAE-MMD and for all remaining models, samples are drawn from a multivariate Gaussian whose mean and covariance are estimated using training set embeddings. For the GMM density estimation, we also utilize the training set embeddings for fitting and validation set embeddings to verify that GMM models are not over fitting to training embeddings. However, due to the very low number of mixture components, we did not encounter overfitting at this step. The GMM parameters are

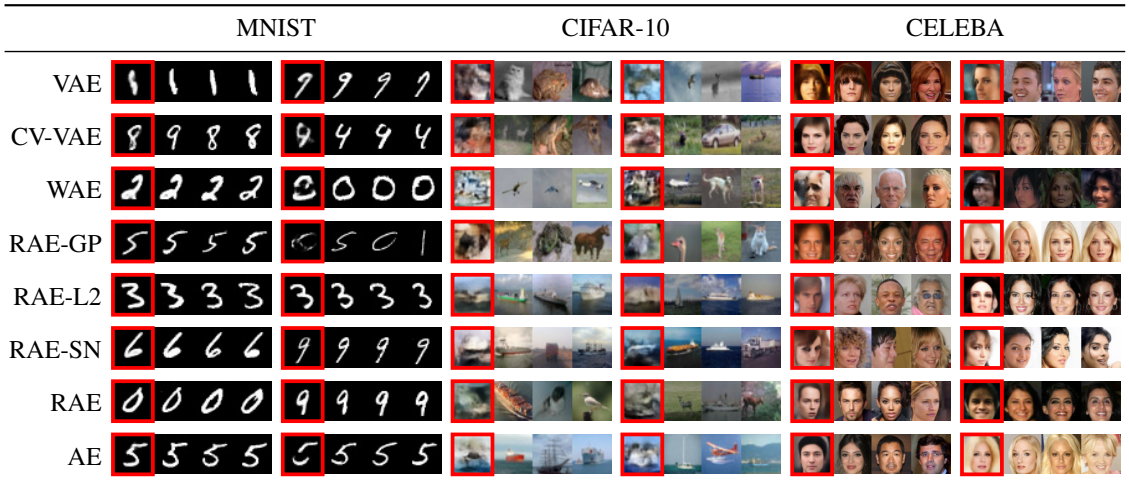


Figure 6.3: Nearest neighbors to generated samples (leftmost image, red box) from training set. It seems that the models have generalized well and fitting only 10 Gaussians to the latent space prevents overfitting.

estimated by running EM for at most 100 iterations.

The network architechtures are shown in Table 6.1. For Figures 6.2 and 6.1, we used smaller networks due to computational limitations and since only relative performance is of interest in these experiments. It should be noted that as a result of this, the absolute values of the reported FID scores in these figures cannot be directly compared with the numbers reported in Section 6.7.

6.8.1 Evaluation

The evaluation of generative models is a nontrivial research question (Theis *et al.*, 2016; Sajjadi *et al.*, 2017; Lucic *et al.*, 2018). Since we are interested in the quality of samples, the ubiquitous Fréchet Inception Distance (FID) (Heusel *et al.*, 2017) is a reasonable choice for comparing different models. More recently, a notion of precision and recall for distributions (PRD) has been proposed (Sajjadi *et al.*, 2018a), separating sample quality from diversity. We choose to report both scores to be able to compare with other less recent works but to also quantitatively measure the precision-recall tradeoffs.

We compute the FID of the reconstructions of random validation samples against the test set to evaluate reconstruction quality. For evaluating generative modeling capabilities, we compute the FID between the test data and randomly drawn samples from a single Gaussian that is either the isotropic $p(\mathbf{z})$ available for VAEs and WAEs, or a single Gaussian fit to $q_\delta(\mathbf{z})$ for CV-VAEs and RAEs. For all models, we also evaluate random samples from a 10-component Gaussian Mixture model (GMM) fit to $q_\delta(\mathbf{z})$. Using only 10 components prevents us from overfitting (which would indeed give good FIDs when compared with the test set). We note that fitting GMMs with up to 100 components, only

	MNIST		CIFAR-10		CelebA	
	$\mathcal{N}$	GMM	$\mathcal{N}$	GMM	$\mathcal{N}$	GMM
VAE	0.96 / 0.92	0.95 / 0.96	0.25 / 0.55	0.37 / 0.56	0.54 / 0.66	0.50 / 0.66
CV-VAE	0.84 / 0.73	0.96 / 0.89	0.31 / 0.64	0.42 / 0.68	0.25 / 0.43	0.32 / 0.55
WAE	0.93 / 0.88	**0.98** / 0.95	0.38 / 0.68	0.51 / **0.81**	0.59 / 0.68	**0.69** / **0.77**
RAE-GP	0.93 / 0.87	0.97 / **0.98**	0.36 / 0.70	0.46 / 0.77	0.38 / 0.55	0.44 / 0.67
RAE-L2	0.92 / 0.87	**0.98** / **0.98**	0.41 / 0.77	**0.57** / **0.81**	0.36 / 0.64	0.44 / 0.65
RAE-SN	0.89 / 0.95	**0.98** / 0.97	0.36 / 0.73	0.52 / **0.81**	0.54 / 0.68	0.55 / 0.74
RAE	0.92 / 0.85	**0.98** / **0.98**	0.45 / 0.73	0.53 / 0.80	0.46 / 0.59	0.52 / 0.69
AE	0.90 / 0.90	**0.98** / 0.97	0.37 / 0.73	0.50 / 0.80	0.45 / 0.66	0.47 / 0.71

Table 6.3: Evaluation of random sample quality by precision / recall (Sajjadi *et al.*, 2018a) (higher numbers are better, best value for each dataset in bold). It is notable that the proposed ex-post density estimation improves not only precision, but also recall throughout the experiment. For example, WAE seems to have a comparably low recall of only 0.88 on MNIST which is raised considerably to 0.95 by fitting a GMM. In all cases, GMM gives the best results. Another interesting point is the low precision but high recall of all models on CIFAR-10 – this is also visible upon inspection of the samples in Figure 6.6.

improved results marginally. Additionally, we provide nearest-neighbors from the training set in Figure 6.3 to show that the models are not overfitting. For interpolations, we report the FID for the furthest interpolation points resulted by applying spherical interpolation to randomly selected validation reconstruction pairs.

We use 10k samples for all FID and PRD evaluations. Reconstruction scores are computed from validation set reconstructions against the respective test set. Interpolation scores are computed by interpolating latent codes of a pair of randomly chosen validation embeddings vs test set samples. The visualized interpolation samples are interpolations between two randomly chosen test set images.

6.8.2 Results

Table 6.2 summarizes our main results. All of the proposed RAE variants are competitive with the VAE and WAE w.r.t. generated image quality in all settings. Sampling RAEs achieve the best FIDs across all datasets when a modest 10-component GMM is employed for ex-post density estimation. Furthermore, even when $\mathcal{N}$ is considered as $q_\delta(\mathbf{z})$, RAEs rank first with the exception of MNIST, though the best FID achieved there by the VAE is very close to the FID of RAE-SN. Table 6.3 reports PRD scores for the same models. We can see that the proposed ex-post density estimation improves not only precision, but also recall throughout the experiment.

Moreover, our best RAE FIDs are lower than the best results reported for VAEs in the large scale comparison of (Lucic *et al.*, 2018), challenging even the best scores reported for GANs. While we are employing a slightly different architecture than theirs, our models underwent only modest finetuning instead of an extensive hyperparameter search. By looking at the differently regularized RAEs, there is no clear winner across all settings as all perform equally well. For practical reasons of implementation simplicity, one may prefer RAE-L2 over the GP and SN variants.

Surprisingly, the implicitly regularized RAE and AE models are shown to be able to score impressive FIDs when $q_\delta(\mathbf{z})$ is fit through GMMs. FIDs for AEs decrease from 58.73 to 10.66 on MNIST and from 127.85 to 45.10 on CelebA – a value close to the state of the art. This is a remarkable result that follows a long series of recent confirmations that neural networks are surprisingly smooth *by design* (Neyshabur *et al.*, 2017). It is also surprising that the lack of an explicitly fixed structure on the latent space of the RAE does not impede interpolation quality. This is further confirmed by the qualitative evaluation on CelebA as reported in Figure 6.4 and for the other datasets in Figures 6.5 and 6.6, where RAE interpolated samples seem sharper than competitors and transitions smoother.

We would like to note that our extensive study confirms and quantifies the effect of the aggregated posterior mismatch as well as the effectivity of our proposed solution to it. Indeed, if we consider the effect of applying ex-post density estimation to each model in Table 6.2, we see that it consistently improves sample quality across all settings considerably. A 10-component GMM trained to fit $\mathbf{Z}$ seems to be enough to half the FID scores from $\sim$20 to $\sim$10 for WAE and RAE models on MNIST and from 116 to 46 on CelebA. This is striking since this very cheap additional step to any VAE-like generative model can be employed to boost the quality of generated samples.

All in all, the results strongly support our conjecture that the simple deterministic RAE framework can challenge the VAE and WAE.

Figure 6.4: Qualitative evaluation of sample quality for VAEs, WAEs and RAEs on CelebA. Left: reconstructed samples (top row is ground truth). Middle: randomly generated samples. Right: interpolations in the latent space between a pair of test images (first and last column). RAE models provide overall sharper samples and reconstructions while interpolating smoothly in the latent space.

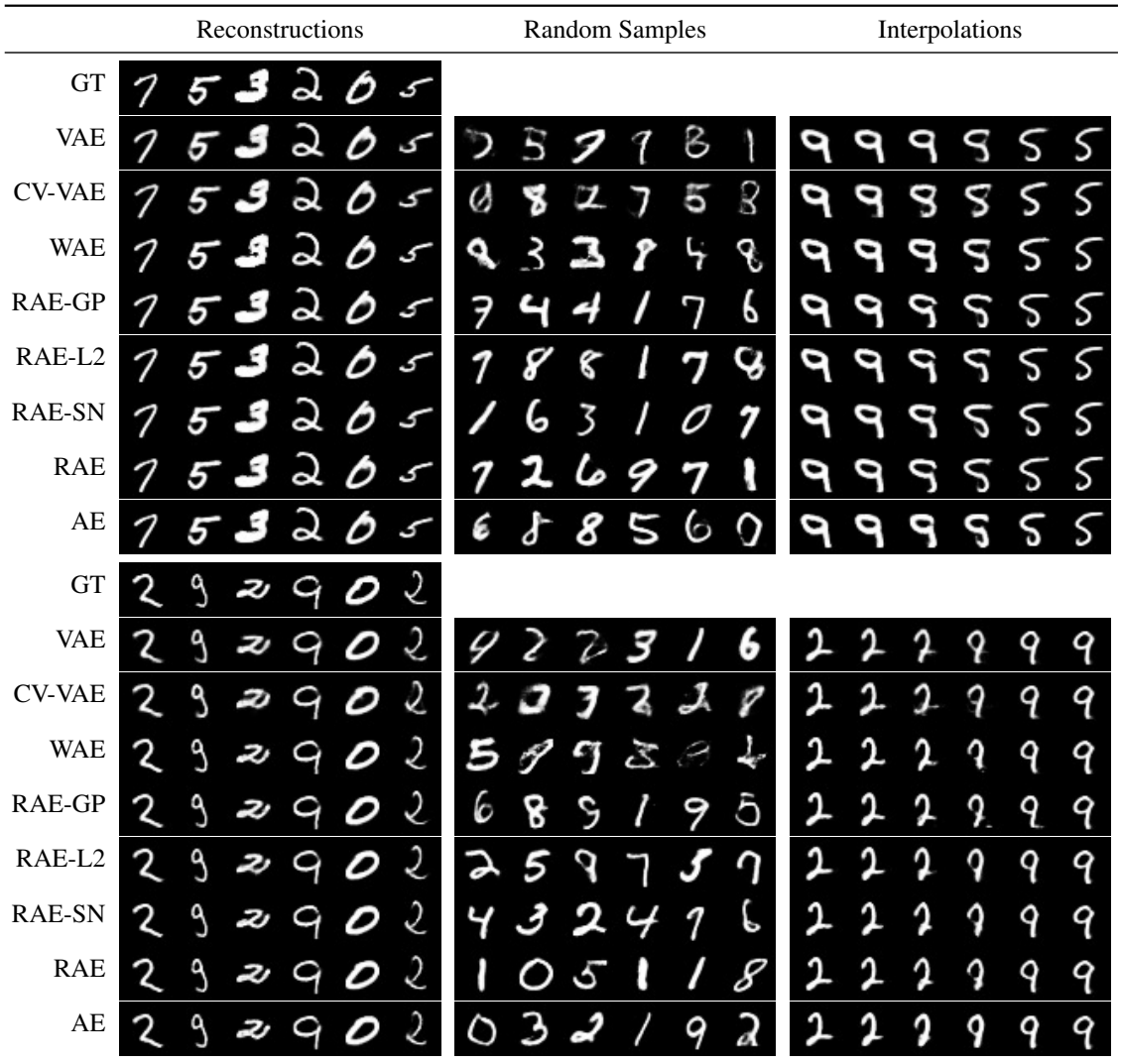


Figure 6.5: Qualitative evaluation for sample quality for VAEs, WAEs and RAEs on MNIST. Left: reconstructed samples (top row is ground truth). Middle: randomly generated samples. Right: spherical interpolations between two images (first and last column).

Figure 6.6: Qualitative evaluation for sample quality for VAEs, WAEs and RAEs on CIFAR-10. Left: reconstructed samples (top row is ground truth). Middle: randomly generated samples. Right: spherical interpolations between two images (first and last column).

6.9 Summary

While the theoretical derivation of the VAE has helped popularize the framework for generative modeling, recent works have started to expose some discrepancies between theory and practice. In this work, viewing sampling in VAEs as noise injection to enforce smoothness has enabled us to distill a deterministic autoencoding framework that is compatible with several regularization techniques to learn a meaningful latent space. We have demonstrated that such a deterministic autoencoding framework can generate comparable or better samples than VAEs, while getting around the practical drawbacks tied to a stochastic framework. Furthermore, we have shown that our solution to fit a simple density estimator such as a Gaussian Mixture Model on the learned latent space is able to consistently improve sample quality both for the proposed RAE framework as well as for the VAE and WAE, acting as a solution for the known mismatch between the prior and the aggregated posterior. The RAE framework opens interesting future research venues such as learning the density estimator in an end-to-end fashion with the autoencoding network and devising more sophisticated autoencoders that can access the full range of recent structural neural network advancements to scale generative modeling without being bound to the VAE's restrictions.

Chapter 7

Conclusions

EnhanceNet (Sajjadi *et al.*, 2017), together with the similar and concurrent approach by Ledig *et al.* (2017), achieved previously unattainable state-of-the-art realism at high upscaling factors up to 4x through the use of novel loss functions that emphasize perceptual fidelity over pixel-wise accuracy of the reconstruction during training. These advances sent waves through the field of super resolution and have since changed it sustainably. The ubiquitous use of the L2 norm as the loss function for image reconstruction tasks has gotten increasingly replaced by more sophisticated loss functions that capture visual fidelity rather than distances in pixel space, particularly for larger upsampling ratios (Wang *et al.*, 2020).

Similarly, low-level similarity metrics for evaluation such as PSNR and structural similarity (SSIM) (Wang *et al.*, 2004) have lost relevance in many applications, being gradually supplanted by perceptual metrics such as the Learned Perceptual Image Patch Similarity (LPIPS) (Zhang *et al.*, 2018) and the Fréchet Inception Distance (FID) (Heusel *et al.*, 2017). As suggested in Chapter 2 Section 2.5.5, both of these more recent works use similarity of pre-trained neural networks, however in the latent space rather than evaluating classification accuracy. Precision and Recall for Distributions (PRD) (Sajjadi *et al.*, 2018a) has turned out to be a practical tool for evaluating the precision and recall of generative models separately, and several follow-up works generalizing and extending the approach have since been published (Kynkäänniemi *et al.*, 2019; Djolonga *et al.*, 2020).

Meanwhile, part of the super resolution field has shifted its focus away from improving metrics, and instead towards improving the applicability of such methods in the wild. EnhanceNet, as the overwhelming majority of works on super resolution at the time, uses HR images and scales them down for training, thereby assuming a Gaussian downscaling kernel. This has the side effect that the input images are often devoid of noise, compression artifacts, blur, and other imperfections. In practice however, these methods should work on imperfect inputs – leading to a more recent spike of interest in learning not only the mapping from LR to HR images, but also its inverse: how to map HR images to (possibly imperfect, but realistic) LR images. A particularly powerful technique for this so-called *blind* image super resolution is the application of GANs to learn the HR to LR mapping (Bell-Kligler *et al.*, 2019).

The fronties of image-generative models have been vastly expanded, with high-resolution image-generative models based on regularized GANs (Mescheder *et al.*, 2018; Liu *et al.*, 2019) contributing to the success of scaling these models to more complex settings. While GANs can still achieve impressive results (Kang *et al.*, 2023), autoregressive methods (Yu *et al.*, 2022) and denoising diffusion (Ho *et al.*, 2020) have taken over, leading in particular to state-of-the-art text to image image generative abilities (Rombach *et al.*, 2022). The most exciting future development lies in extending today's approaches from image to video format, which brings not only the obvious challenge of significantly larger and harder to amass data, but also that of temporal consistency. It remains to be seen how far approaches similar to FRVSR (Sajjadi *et al.*, 2018b) can be taken in this domain.

Exhaustive list of co-authored papers during the PhD studies

- Sajjadi, Alamgir, and von Luxburg (2016b). Peer Grading in a Course on Algorithms and Data Structures: Machine Learning Algorithms do not Improve over Simple Baselines. Learning at Scale (L@S) 2016.
- Sajjadi, Köhler, Hirsch, and Schölkopf (2016a). Depth Estimation Through a Generative Model of Light Field Synthesis. German Conference on Pattern Recognition (GCPR) 2016.
- Sajjadi, Schölkopf, and Hirsch (2017). EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis. International Conference on Computer Vision (ICCV) 2017 (oral).
- Sajjadi, Vemulapalli, and Brown (2018b). Frame-Recurrent Video Super-Resolution. Computer Vision and Pattern Recognition (CVPR) 2018.
- Sajjadi, Parascandolo, Mehrjou, and Schölkopf (2018c). Tempered Adversarial Networks. International Conference on Machine Learning (ICML) 2018 (oral).
- Kim, Sajjadi, Hirsch, and Schölkopf (2018). Spatio-Temporal Transformer Network for Video Restoration. European Conference on Computer Vision (ECCV) 2018.
- Perez-Pellitero, Sajjadi, Hirsch, and Schölkopf (2018). Photorealistic Video Super Resolution. European Conference on Computer Vision (ECCV) 2018 Workshop PIRM.
- Sajjadi, Bachem, Lucic, Bousquet, and Gelly (2018a). Assessing Generative Models via Precision and Recall. Neural Information Processing Systems (NeurIPS) 2018.
- Ghosh*, Sajjadi*, Vergari, Black, and Schölkopf (2019). From Variational to Deterministic Autoencoders. *equal contribution. International Conference on Learning Representations (ICLR) 2019.

List of Tables

List of Figures

Bibliography

Abadi et. al., M. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.

Agustsson, E. and Timofte, R. (2017). NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In *Computer Vision and Pattern Recognition (CVPR) Workshop*.

Alemi, A., Poole, B., Fischer, I., Dillon, J., Saurous, R. A., and Murphy, K. (2018). Fixing a Broken ELBO. In *International Conference on Machine Learning (ICML)*.

An, G. (1996). The Effects of Adding Noise During Backpropagation Training on a Generalization Performance. In *Neural computation*.

Arjovsky, M. and Bottou, L. (2017). Towards Principled Methods for Training Generative Adversarial Networks. In *International Conference on Machine Learning (ICML)*.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning (ICML)*.

Bachem, O., Lucic, M., Hassani, S. H., and Krause, A. (2016). Approximate K-Means++ in Sublinear Time. In *AAAI Conference on Artificial Intelligence*.

Ballé, J., Laparra, V., and Simoncelli, E. P. (2017). End-to-End Optimized Image Compression. In *International Conference on Learning Representations (ICLR)*.

Barratt, S. and Sharma, R. (2018). A Note on the Inception Score. *Workshop on Theoretical Foundations and Applications of Deep Generative Models, ICML 2018*.

Bauer, M. and Mnih, A. (2019). Resampled Priors for Variational Autoencoders. In *AAAI Conference on Artificial Intelligence*.

Belekos, S. P., Galatsanos, N. P., and Katsaggelos, A. K. (2010). Maximum a Posteriori Video Super-Resolution Using a new Multichannel Image Prior. *IEEE Transactions on Image Processing (TIP)*.

Bell-Kligler, S., Shocher, A., and Irani, M. (2019). Blind Super-Resolution Kernel Estimation using an Internal-GAN. *Neural Information Processing Systems (NeurIPS)*.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum Learning. In *International Conference on Machine Learning (ICML)*.

Bengio, Y., Yao, L., Alain, G., and Vincent, P. (2013). Generalized Denoising Auto-Encoders as Generative Models. In *Neural Information Processing Systems (NeurIPS)*.

Berthelot, D., Schumm, T., and Metz, L. (2017). BEGAN: Boundary Equilibrium Generative Adversarial Networks. *arXiv:1703.10717*.

Bevilacqua, M., Roumy, A., Guillemot, C., and Morel, M.-L. A. (2012). Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. In *British Machine Vision Conference (BMVC)*.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Bikowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. (2018). Demystifying MMD GANs. In *International Conference on Learning Representations (ICLR)*.

Borji, A. (2018). Pros and Cons of GAN Evaluation Measures. *Computer Vision and Image Understanding*.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2016). Generating Sentences from a Continuous Space. In *Conference on Natural Language Learning (CoNLL)*.

Brock, A., Donahue, J., and Simonyan, K. (2019). Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *International Conference on Learning Representations (ICLR)*.

Bruna, J., Sprechmann, P., and LeCun, Y. (2016). Super-Resolution with Deep Convolutional Sufficient Statistics. In *International Conference on Learning Representations (ICLR)*.

Burda, Y., Grosse, R., and Salakhutdinov, R. (2015). Importance Weighted Autoencoders. *arXiv:1509.00519*.

Caballero, J., Ledig, C., Aitken, A., Acosta, A., and Totz (2017). Real-Time Video Super-Resolution with Spatio-Temporal Networks and Motion Compensation. In *Computer Vision and Pattern Recognition (CVPR)*.

Chang, H., Yeung, D.-Y., and Xiong, Y. (2004). Super-Resolution Through Neighbor Embedding. In *Computer Vision and Pattern Recognition (CVPR)*.

Chen, D., Liao, J., Yuan, L., Yu, N., and Hua, G. (2017a). Coherent Online Video Style Transfer. In *International Conference on Computer Vision (ICCV)*.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Neural Information Processing Systems (NeurIPS)*.

Chen, X., Kingma, D. P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I., and Abbeel, P. (2017b). Variational Lossy Autoencoder. In *International Conference on Learning Representations (ICLR)*.

Cífka, O., Severyn, A., Alfonseca, E., and Filippova, K. (2018). Eval All, Trust a Few, do Wrong to None: Comparing Sentence Generation Models. *arXiv:1804.07972*.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. *Empirical Methods in Natural Language Processing (EMNLP)*.

Cowles, M. K. and Carlin, B. P. (1996). Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review. In *Journal of the American Statistical Association*.

Dahl, R. (2016). ResNet in TensorFlow. `https://github.com/ry/tensorflow-resnet`. (visited on November 10, 2016).

Dai, B. and Wipf, D. (2019). Diagnosing and Enhancing VAE Models. In *International Conference on Learning Representations (ICLR)*.

Denton, E. L., Chintala, S., Fergus, R., *et al.* (2015). Deep Generative Image Models Using a Laplacian Pyramid of Adversarial Networks. In *Neural Information Processing Systems (NeurIPS)*.

Djolonga, J., Lucic, M., Cuturi, M., Bachem, O., Bousquet, O., and Gelly, S. (2020). Precision-Recall Curves using Information Divergence Frontiers. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Dong, C., Loy, C. C., He, K., and Tang, X. (2014). Learning a Deep Convolutional Network for Image Super-Resolution. In *European Conference on Computer Vision (ECCV)*.

Dong, C., Loy, C. C., and Tang, X. (2016). Accelerating the Super-Resolution Convolutional Neural Network. In *European Conference on Computer Vision (ECCV)*.

Dosovitskiy, A. and Brox, T. (2016). Generating Images with Perceptual Similarity Metrics based on Deep Networks. In *Neural Information Processing Systems (NeurIPS)*.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. (2015). Flownet: Learning Optical Flow with Convolutional Networks. In *International Conference on Computer Vision (ICCV)*.

Drulea, M. and Nedevschi, S. (2011). Total Variation Regularization of Local-Global Optical Flow. In *International Conference on Intelligent Transportation Systems (ITSC)*.

Duchon, C. E. (1979). Lanczos Filtering in One and Two Dimensions. *Journal of Applied Meteorology*.

Farsiu, S., Robinson, M. D., Elad, M., and Milanfar, P. (2004). Fast and Robust Multiframe Super Resolution. *IEEE Transactions on Image Processing (TIP)*.

Farsiu, S., Elad, M., and Milanfar, P. (2006). Video-to-Video Dynamic Super-Resolution for Grayscale and Color Sequences. *EURASIP Journal on Applied Signal Processing*.

Fookes, C., Lin, F., Chandran, V., and Sridharan, S. (2012). Evaluation of Image Resolution and Super-Resolution on Face Recognition Performance. *Journal of Visual Communication and Image Representation*.

Freedman, G. and Fattal, R. (2011). Image and Video Upscaling from Local Self-Examples. *ACM Transactions on Graphics (TOG)*.

Freeman, W. T., Jones, T. R., and Pasztor, E. C. (2002). Example-Based Super-Resolution. *IEEE Computer Graphics and Applications (CG&A)*.

Gatys, L., Ecker, A. S., and Bethge, M. (2015). Texture Synthesis Using Convolutional Neural Networks. In *Neural Information Processing Systems (NeurIPS)*.

Gatys, L. A., Ecker, A. S., and Bethge, M. (2016). Image Style Transfer Using Convolutional Neural Networks. In *Computer Vision and Pattern Recognition (CVPR)*.

Ghosh*, P., Sajjadi*, M. S. M., Vergari, A., Black, M., and Schölkopf, B. (2019). From Variational to Deterministic Autoencoders. In *International Conference on Learning Representations (ICLR)*.

Ghosh, P., Losalka, A., and Black, M. J. (2019). Resisting Adversarial Attacks Using Gaussian Mixture Variational Autoencoders. In *AAAI Conference on Artificial Intelligence*.

Glasner, D., Bagon, S., and Irani, M. (2009). Super-Resolution from a Single Image. In *International Conference on Computer Vision (ICCV)*.

Glorot, X. and Bengio, Y. (2010). Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *AAAI Conference on Artificial Intelligence*.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. (2018). Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. In *ACS Central Science*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Nets. In *Neural Information Processing Systems (NeurIPS)*.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Gulcehre, C., Moczulski, M., Visin, F., and Bengio, Y. (2017). Mollifying Networks. *International Conference on Learning Representations (ICLR)*.

Gulrajani, I. (2017). Code for Reproducing Experiments in Improved Training of Wasserstein GANs. `https://github.com/igul222/improved_wgan_training`. (Latest commit from June 22, 2017).

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved Training of Wasserstein GANs. In *Neural Information Processing Systems (NeurIPS)*.

Gupta, A., Johnson, J., Alahi, A., and Fei-Fei, L. (2017). Characterizing and Improving Stability in Neural Style Transfer. In *International Conference on Computer Vision (ICCV)*.

HaCohen, Y., Fattal, R., and Lischinski, D. (2010). Image Upsampling via Texture Hallucination. In *IEEE International Conference on Computational Photography (ICCP)*.

Ham, C., Raj, A., Cartillier, V., and Essa, I. (2018). Variational Image Inpainting. In *Bayesian Deep Learning Workshop, NeurIPS*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Computer Vision and Pattern Recognition (CVPR)*.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Neural Information Processing Systems (NeurIPS)*.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). Beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations (ICLR)*.

Hinton, G., Srivastava, N., and Swersky, K. (2012). Neural Networks for Machine Learning. Overview of Mini-Batch Gradient Descent.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Neural Information Processing Systems (NeurIPS)*.

Hoffman, M. D. and Johnson, M. J. (2016). ELBO Surgery: Yet Another Way to Carve up the Variational Evidence Lower Bound. In *Workshop in Advances in Approximate Bayesian Inference, NeurIPS*.

Hradiš, M., Kotera, J., Zemcík, P., and Šroubek, F. (2015). Convolutional Neural Networks for Direct Text Deblurring. In *British Machine Vision Conference (BMVC)*.

Huang, H., He, R., Sun, Z., Tan, T., *et al.* (2018). Introvae: Introspective Variational Autoencoders for Photographic Image Synthesis. In *Neural Information Processing Systems (NeurIPS)*.

Huang, J.-B., Singh, A., and Ahuja, N. (2015a). Single Image Super-Resolution from Transformed Self-Exemplars. In *Computer Vision and Pattern Recognition (CVPR)*.

Huang, Y., Wang, W., and Wang, L. (2015b). Bidirectional Recurrent Convolutional Networks for Multi-Frame Super-Resolution. In *Neural Information Processing Systems (NeurIPS)*.

Huszár, F. (2015). How (not) to Train your Generative Model: Scheduled Sampling, Likelihood, Adversary? *arXiv:1511.05101*.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. (2017). Flownet 2.0: Evolution of Optical Flow Estimation with Deep Networks. *Computer Vision and Pattern Recognition (CVPR)*.

Im, D. J., Ma, H., Taylor, G., and Branson, K. (2018). Quantitatively Evaluating GANs with Divergences Proposed for Training. In *International Conference on Learning Representations (ICLR)*.

Ioffe, S. and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning (ICML)*.

Irani, M. and Peleg, S. (1991). Improving Resolution by Image Registration. *Graphical Models and Image Processing (CVGIP)*.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). Spatial Transformer Networks. In *Neural Information Processing Systems (NeurIPS)*.

Jin, W., Barzilay, R., and Jaakkola, T. (2018). Junction Tree Variational Autoencoder for Molecular Graph Generation. *International Conference on Machine Learning (ICML)*.

Johnson, J., Alahi, A., and Fei-Fei, L. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *European Conference on Computer Vision (ECCV)*.

Kang, M., Zhu, J.-Y., Zhang, R., Park, J., Shechtman, E., Paris, S., and Park, T. (2023). Scaling up GANs for Text-to-Image Synthesis. In *Computer Vision and Pattern Recognition (CVPR)*.

Kappeler, A., Yoo, S., Dai, Q., and Katsaggelos, A. K. (2016). Video Super-Resolution with Convolutional Neural Networks. In *IEEE Transactions on Computational Imaging (TCI)*.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive Growing of GANs for Improved Quality, Stability, and Variation. *International Conference on Learning Representations (ICLR)*.

Kim, J., Kwon Lee, J., and Mu Lee, K. (2016a). Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In *Computer Vision and Pattern Recognition (CVPR)*.

Kim, J., Kwon Lee, J., and Mu Lee, K. (2016b). Deeply-Recursive Convolutional Network for Image Super-Resolution. In *Computer Vision and Pattern Recognition (CVPR)*.

Kim, K. I. and Kwon, Y. (2010). Single-Image Super-Resolution Using Sparse Regression and Natural Image Prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Kim, T. H., Lee, K. M., Schölkopf, B., and Hirsch, M. (2017). Online Video Deblurring via Dynamic Temporal Blending Network. In *International Conference on Computer Vision (ICCV)*.

Kim, T. H., Sajjadi, M. S. M., Hirsch, M., and Schölkopf, B. (2018). Spatio-Temporal Transformer Network for Video Restoration. In *European Conference on Computer Vision (ECCV)*.

Kingma, D. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*.

Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes. In *International Conference on Learning Representations (ICLR)*.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improving Variational Inference with Inverse Autoregressive Flow. In *Neural Information Processing Systems (NeurIPS)*.

Krizhevsky, A. and Hinton, G. (2009). Learning Multiple Layers of Features from Tiny Images.

Kurach, K., Lucic, M., Zhai, X., Michalski, M., and Gelly, S. (2019). A Large-Scale Study on Regularization and Normalization in GANs. *International Conference on Machine Learning (ICML)*.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. (2019). Improved Precision and Recall Metric for Assessing Generative Models. *Neural Information Processing Systems (NeurIPS)*.

Lai, W.-S., Huang, J.-B., Ahuja, N., and Yang, M.-H. (2017). Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution. In *Computer Vision and Pattern Recognition (CVPR)*.

Laparra, V., Ballé, J., Berardino, A., and Simoncelli, E. P. (2016). Perceptual Image Quality Assessment Using a Normalized Laplacian Pyramid. *Journal of Electronic Imaging*.

Larochelle, H. and Murray, I. (2011). The Neural Autoregressive Distribution Estimator. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *Computer Vision and Pattern Recognition (CVPR)*.

Lin, F., Fookes, C., Chandran, V., and Sridharan, S. (2007). Super-Resolved Faces for Improved Face Recognition from Surveillance Video. In *International Conference on Biometrics (ICB)*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*.

Liu, C. and Sun, D. (2011). A Bayesian Approach to Adaptive Video Super Resolution. In *Computer Vision and Pattern Recognition (CVPR)*.

Liu, D., Wang, Z., Fan, Y., Liu, X., Wang, Z., Chang, S., and Huang, T. (2017). Robust Video Super-Resolution with Learned Temporal Dynamics. In *Computer Vision and Pattern Recognition (CVPR)*.

Liu, K., Tang, W., Zhou, F., and Qiu, G. (2019). Spectral Regularization for Combating Mode Collapse in GANs. In *Computer Vision and Pattern Recognition (CVPR)*.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015a). Deep Learning Face Attributes in the Wild. In *International Conference on Computer Vision (ICCV)*.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015b). Deep Learning Face Attributes in the Wild. In *International Conference on Computer Vision (ICCV)*.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. In *Computer Vision and Pattern Recognition (CVPR)*.

Lopez-Paz, D. and Oquab, M. (2016). Revisiting Classifier Two-Sample Tests. In *International Conference on Learning Representations (ICLR)*.

Lu, X., Yuan, H., Yan, P., Yuan, Y., and Li, X. (2012). Geometry Constrained Sparse Coding for Single Image Super-Resolution. In *Computer Vision and Pattern Recognition (CVPR)*.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. (2018). Are GANs Created Equal? A Large-Scale Study. In *Neural Information Processing Systems (NeurIPS)*.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *International Conference on Machine Learning (ICML)*.

Machrisaa (2016). VGG19 and VGG16 on Tensorflow. `https://github.com/machrisaa/tensorflow-vgg`. (visited on June 6, 2016).

Makansi, O., Ilg, E., and Brox, T. (2017). End-to-End Learning of Video Super-Resolution with Motion Compensation. In *German Conference on Pattern Recognition (GCPR)*.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2016). Adversarial Autoencoders. In *International Conference on Learning Representations (ICLR)*.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Smolley, S. P. (2017). Least Squares Generative Adversarial Networks. In *International Conference on Computer Vision (ICCV)*.

Mescheder, L., Geiger, A., and Nowozin, S. (2018). Which Training Methods for GANs do Actually Converge? In *International Conference on Machine Learning (ICML)*.

Milanfar, P. (2010). *Super-Resolution Imaging*. CRC Press.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations (ICLR)*.

Namboodiri, V. P., De Smet, V., and Van Gool, L. (2011). Systematic Evaluation of Super-Resolution Using Classification. In *IEEE International Conference on Visual Communications and Image Processing (VCIP)*.

Nasrollahi, K. and Moeslund, T. B. (2014). Super-Resolution: A Comprehensive Survey. *Machine Vision and Applications*.

Neyshabur, B., Tomioka, R., Salakhutdinov, R., and Srebro, N. (2017). Geometry of Optimization and Implicit Regularization in Deep Learning. *arXiv:1705.03071*.

Odena, A., Dumoulin, V., and Olah, C. (2016). Deconvolution and Checkerboard Artifacts. http://distill.pub/2016/deconv-checkerboard/.

Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context Encoders: Feature Learning by Inpainting. In *Computer Vision and Pattern Recognition (CVPR)*.

Perez-Pellitero, E., Salvador, J., Ruiz-Hidalgo, J., and Rosenhahn, B. (2016). PSyCo: Manifold Span Reduction for Super Resolution. In *Computer Vision and Pattern Recognition (CVPR)*.

Perez-Pellitero, E., Sajjadi, M. S. M., Hirsch, M., and Schölkopf, B. (2018). Photorealistic Video Super Resolution. In *European Conference on Computer Vision (ECCV) Workshop PIRM*.

Radford, A., Metz, L., and Chintala, S. (2016a). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *International Conference on Learning Representations (ICLR)*.

Radford, A., Metz, L., and Chintala, S. (2016b). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *International Conference on Learning Representations (ICLR)*.

Ranjan, A. and Black, M. J. (2017). Optical Flow Estimation Using a Spatial Pyramid Network. *Computer Vision and Pattern Recognition (CVPR)*.

Rezende, D. J. and Viola, F. (2018). Taming VAEs. *arXiv:1810.00597*.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *International Conference on Machine Learning (ICML)*.

Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. (2011). Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. In *International Conference on Machine Learning (ICML)*.

Rifai, S., Bengio, Y., Dauphin, Y., and Vincent, P. (2012). A Generative Process for Sampling Contractive Auto-Encoders. In *International Conference on Machine Learning (ICML)*.

Romano, Y., Isidoro, J., and Milanfar, P. (2016). RAISR: Rapid and Accurate Image Super Resolution. *IEEE Transactions on Computational Imaging (TCI)*.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-Resolution Image Synthesis with Latent Diffusion Models. *Computer Vision and Pattern Recognition (CVPR)*.

Rosca, M., Lakshminarayanan, B., and Mohamed, S. (2018). Distribution Matching in Variational Inference. *arXiv:1802.06847*.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear Total Variation Based Noise Removal Algorithms. *Physica D: Nonlinear Phenomena*.

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. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*.

Sajjadi, M. S. M., Köhler, R., Hirsch, M., and Schölkopf, B. (2016a). Depth Estimation Through a Generative Model of Light Field Synthesis. In *German Conference on Pattern Recognition (GCPR)*.

Sajjadi, M. S. M., Alamgir, M., and von Luxburg, U. (2016b). Peer Grading in a Course on Algorithms and Data Structures: Machine Learning Algorithms do not Improve over Simple Baselines. In *Proceedings of the 3rd ACM conference on Learning at Scale*.

Sajjadi, M. S. M., Schölkopf, B., and Hirsch, M. (2017). EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis. In *International Conference on Computer Vision (ICCV)*.

Sajjadi, M. S. M., Bachem, O., Lucic, M., Bousquet, O., and Gelly, S. (2018a). Assessing Generative Models via Precision and Recall. In *Neural Information Processing Systems (NeurIPS)*.

Sajjadi, M. S. M., Vemulapalli, R., and Brown, M. (2018b). Frame-Recurrent Video Super-Resolution. In *Computer Vision and Pattern Recognition (CVPR)*.

Sajjadi, M. S. M., Parascandolo, G., Mehrjou, A., and Schölkopf, B. (2018c). Tempered Adversarial Networks. In *International Conference on Machine Learning (ICML)*.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016a). Improved Techniques for Training GANs. *Neural Information Processing Systems (NeurIPS)*.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016b). Improved Techniques for Training GANs. In *Neural Information Processing Systems (NeurIPS)*.

Schulter, S., Leistner, C., and Bischof, H. (2015). Fast and Accurate Image Upscaling with Super-Resolution Forests. In *Computer Vision and Pattern Recognition (CVPR)*.

Sculley, D. (2010). Web-Scale K-Means Clustering. In *International Conference on World Wide Web (WWW)*.

Severyn, A., Barth, E., and Semeniuta, S. (2017). A Hybrid Convolutional Variational Autoencoder for Text Generation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., and Wang, Z. (2016a). Real-time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In *Computer Vision and Pattern Recognition (CVPR)*.

Shi, Y., Wang, K., Xu, L., and Lin, L. (2016b). Local-and Holistic-Structure Preserving Image Super Resolution via Deep Joint Component Learning. In *IEEE International Conference on Multimedia and Expo (ICME)*.

Sietsma, J. and Dow, R. J. (1991). Creating Artificial Neural Networks That Generalize. In *Neural Networks*.

Simonyan, K. and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*.

Sohn, K., Lee, H., and Yan, X. (2015). Learning Structured Output Representation Using Deep Conditional Generative Models. In *Neural Information Processing Systems (NeurIPS)*.

Sønderby, C. K., Caballero, J., Theis, L., Shi, W., and Huszár, F. (2017). Amortised MAP Inference for Image Super-Resolution. In *International Conference on Learning Representations (ICLR)*.

Srivastava, A., Valkoz, L., Russell, C., Gutmann, M. U., and Sutton, C. (2017). VEEGAN: Reducing Mode Collapse in GANs Using Implicit Variational Learning. In *Neural Information Processing Systems (NeurIPS)*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In *Journal of Machine Learning Research (JMLR)*.

Sun, L. and Hays, J. (2012). Super-Resolution from Internet-Scale Scene Matching. In *IEEE International Conference on Computational Photography (ICCP)*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In *Computer Vision and Pattern Recognition (CVPR)*.

Tai, Y., Yang, J., and Liu, X. (2017). Image Super-Resolution via Deep Recursive Residual Network. In *Computer Vision and Pattern Recognition (CVPR)*.

Tai, Y.-W., Liu, S., Brown, M. S., and Lin, S. (2010). Super Resolution Using Edge Prior and Single Image Detail Synthesis. In *Computer Vision and Pattern Recognition (CVPR)*.

Takeda, H., Milanfar, P., Protter, M., and Elad, M. (2009). Super-Resolution without Explicit Subpixel Motion Estimation. *IEEE Transactions on Image Processing (TIP)*.

Tao, X., Gao, H., Liao, R., Wang, J., and Jia, J. (2017). Detail-revealing Deep Video Super-Resolution. In *International Conference on Computer Vision (ICCV)*.

Theis, L., Oord, A. v. d., and Bethge, M. (2016). A Note on the Evaluation of Generative Models. In *International Conference on Learning Representations (ICLR)*.

Tikhonov, A. N. and Arsenin, V. I. (1977). *Solutions of Ill Posed Problems*. Vh Winston.

Timofte, R., De Smet, V., and Van Gool, L. (2014). A+: Adjusted Anchored Neighborhood Regression for Fast Super-Resolution. In *Asian Conference on Computer Vision (ACCV)*.

Timofte, R., Rothe, R., and Van Gool, L. (2016). Seven Ways to Improve Example-Based Single Image Super Resolution. In *Computer Vision and Pattern Recognition (CVPR)*.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. (2017). Wasserstein Auto-Encoders. In *International Conference on Learning Representations (ICLR)*.

Tomczak, J. and Welling, M. (2018). VAE with a VampPrior. In *AAAI Conference on Artificial Intelligence*.

Tucker, G., Mnih, A., Maddison, C. J., Lawson, J., and Sohl-Dickstein, J. (2017). REBAR: Low-Variance, Unbiased Gradient Estimates for Discrete Latent Variable Models. In *Neural Information Processing Systems (NeurIPS)*.

Ulyanov, D., Lebedev, V., Vedaldi, A., and Lempitsky, V. (2016). Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. In *International Conference on Machine Learning (ICML)*.

Van den Oord, A., Vinyals, O., *et al.* (2017). Neural Discrete Representation Learning. In *Neural Information Processing Systems (NeurIPS)*.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and Composing Robust Features With Denoising Autoencoders. In *International Conference on Machine Learning (ICML)*.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing (TIP)*.

Wang, Z., Chen, J., and Hoi, S. C. (2020). Deep Learning for Image Super-Resolution: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Williams, A., Nangia, N., and Bowman, S. R. (2018). A Broad-Coverage Challenge Corpus for Sentence Understanding Through Inference. *North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*.

Wu, Y., Burda, Y., Salakhutdinov, R., and Grosse, R. (2017). On the Quantitative Analysis of Decoder-Based Generative Models. In *International Conference on Learning Representations (ICLR)*.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv:1708.07747*.

Xiao, L., Wang, J., Heidrich, W., and Hirsch, M. (2016). Learning High-Order Filters for Efficient Blind Deconvolution of Document Photographs. In *European Conference on Computer Vision (ECCV)*.

Yang, C.-Y., Huang, J.-B., and Yang, M.-H. (2010a). Exploiting Self-Similarities for Single Frame Super-Resolution. In *Asian Conference on Computer Vision (ACCV)*.

Yang, C.-Y., Ma, C., and Yang, M.-H. (2014). Single-Image Super-Resolution: A Benchmark. In *European Conference on Computer Vision (ECCV)*.

Yang, J., Wright, J., Huang, T. S., and Ma, Y. (2010b). Image Super-Resolution via Sparse Representation. *IEEE Transactions on Image Processing (TIP)*.

Yang, J., Wang, Z., Lin, Z., Cohen, S., and Huang, T. (2012). Coupled Dictionary Training for Image Super-Resolution. *IEEE Transactions on Image Processing (TIP)*.

Yang, J., Lin, Z., and Cohen, S. (2013). Fast Image Super-Resolution based on In-Place Example Regression. In *Computer Vision and Pattern Recognition (CVPR)*.

Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., *et al.* (2022). Scaling Autoregressive Models for Content-Rich Text-to-Image Generation. *arXiv:2206.10789*.

Yu, X. and Porikli, F. (2016). Ultra-Resolving Face Images by Discriminative Generative Networks. In *European Conference on Computer Vision (ECCV)*.

Yue, H., Sun, X., Yang, J., and Wu, F. (2013). Landmark Image Super-Resolution by retrieving web Images. *IEEE Transactions on Image Processing (TIP)*.

Zagoruyko, S. and Komodakis, N. (2016). Wide Residual Networks. In *British Machine Vision Conference (BMVC)*.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017a). Understanding Deep Learning Requires Rethinking Generalization. In *International Conference on Learning Representations (ICLR)*.

Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X., and Metaxas, D. (2017b). StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In *International Conference on Computer Vision (ICCV)*.

Zhang, K., Gao, X., Tao, D., and Li, X. (2012). Multi-Scale Dictionary for Single Image Super-Resolution. In *Computer Vision and Pattern Recognition (CVPR)*.

Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful Image Colorization. In *European Conference on Computer Vision (ECCV)*.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Computer Vision and Pattern Recognition (CVPR)*.

Zhao, J., Mathieu, M., and LeCun, Y. (2017a). Energy-Based Generative Adversarial Network. *International Conference on Learning Representations (ICLR)*.

Zhao, S., Song, J., and Ermon, S. (2017b). Towards Deeper Understanding of Variational Autoencoding Models. *arXiv:1702.08658*.

Zhu, J.-Y., Krähenbühl, P., Shechtman, E., and Efros, A. A. (2016). Generative Visual Manipulation on the Natural Image Manifold. In *European Conference on Computer Vision (ECCV)*.