Optimizing the Latent Space of Deep Generative Models

Dissertation
to obtain the degree of
Doctor of Natural Sciences
(Dr. rer. nat.)

submitted by
Amrutha Saseendran

submitted to the School of Science and Technology
of the University of Siegen
Siegen 2023

Supervisor and first appraiser
Prof. Dr.-Ing. Margret Keuper
University of Siegen

Second appraiser
Prof. Dr. Michael Möller
University of Siegen

Dedicated to Emil

Declaration

I hereby declare in lieu of an oath that I have drawn up the present work without any undue assistance by third parties and without using any aids other than the ones specified. The data and concepts taken, either directly or indirectly, from any other sources have been marked, indicating the source.
The work has not been submitted to any other examining authority neither in Germany nor abroad and neither in the same nor in any similar form.
Use of the services of any PhD mediation institute or of any similar organisation has not been made.
Any family relationship, first-degree relationship, marriage, civil partnership or cohabitation to the proposed members of the PhD Commission do not exist.

submitted by
Amrutha Saseendran
December 2023

Abstract

Deep generative models are powerful machine learning models used to model high-dimensional complex data distributions. The rich and semantically expressive latent representations learned by these models are used for various downstream applications in computer vision and natural language processing. It is evident that the effectiveness of the generative techniques highly depends on the quality of the learned representations. Hence in this dissertation, we focus on improving the desirable properties of the learned latent space of two popular deep generative models, Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs). Specifically, we focus on properties such as generalizability, controllability, smoothness, and adversarial robustness.

In the first technical contribution we present in this work, we focus on improving the controllability of latent representations in GANs to generate high-quality images. To be precise, we propose a method to control the content of the generated images solely based on the defined number of objects from multiple classes and introduce a state-of-the-art conditioned adversarial network. We also introduce a real-world count-based dataset called CityCount to validate our results in challenging scenarios.

Next, we explore the learned representations of VAEs and some of the practical limitations associated with them. To this end, we propose a simple, novel, and end-to-end trainable deterministic autoencoding method that efficiently structures the latent space of the model during training and leverages the capacity of expressive multimodal latent distributions. We demonstrate the potential of the proposed method for modeling both continuous and discrete data structures. Finally, we investigate the adversarial robustness of the learned representations in VAEs. One of the major limitations in existing robust VAE models is the trade-off between the quality of image generation and the robustness achieved. We show that the learned representations in the proposed regularized deterministic autoencoders with a comparatively cheap adversarial learning scheme exhibit superior robustness to adversarial attacks without compromising the quality of image generation.

Zusammenfassung

Tiefe generative Modelle sind leistungsstarke maschinelle Lernmodelle, die zur Modellierung hochdimensionaler komplexer Datenverteilungen verwendet werden. Die reichhaltigen und semantisch aussagekräftigen latenten Repräsentationen, die von diesen Modellen erlernt werden, werden für verschiedene Anwendungen in der Computer Vision und der Verarbeitung natürlicher Sprache verwendet. Es ist offensichtlich, dass die Effektivität der generativen Techniken in hohem Maße von der Qualität der erlernten Repräsentationen abhängt. Daher konzentrieren wir uns in dieser Dissertation auf die Verbesserung der Eigenschaften des erlernten latenten Raums von zwei weit verbreiteten tiefen generativen Modellen, Generative Adversarial Networks (GANs) und Variational Autoencoders (VAEs). Insbesondere konzentrieren wir uns auf Eigenschaften wie Generalisierungsfähigkeit, Kontrollierbarkeit, Glattheit und Widerstandsfähigkeit gegenüber widrigen Umständen.

Im ersten technischen Beitrag, den wir in dieser Arbeit vorstellen, konzentrieren wir uns auf die Verbesserung der Kontrollierbarkeit latenter Darstellungen in GANs, um qualitativ hochwertige Bilder zu erzeugen. Um genau zu sein, schlagen wir eine Methode vor, um den Inhalt der generierten Bilder allein auf der Grundlage der definierten Anzahl von Objekten aus mehreren Klassen zu kontrollieren, und führen ein modernes konditioniertes adversarisches Netzwerk ein. Außerdem stellen wir einen realen zählbasierten Datensatz namens CityCount vor, um unsere Ergebnisse in anspruchsvollen Szenarien zu validieren.

Als nächstes untersuchen wir die erlernten Darstellungen von VAEs und einige der damit verbundenen praktischen Einschränkungen. Zu diesem Zweck schlagen wir eine einfache, neuartige und durchgängig trainierbare deterministische Autocodierungsmethode vor, die den latenten Raum des Modells während des Trainings effizient strukturiert und die Kapazität ausdrucksstarker multimodaler latenter Verteilungen nutzt. Wir demonstrieren das Potenzial der vorgeschlagenen Methode für die Modellierung sowohl kontinuierlicher als auch diskreter Datenstrukturen. Schließlich untersuchen wir die Robustheit der erlernten Repräsentationen in VAEs gegenüber nachteiligen Einflüssen. Eine der größten Einschränkungen bei bestehenden robusten VAE-Modellen ist der Kompromiss zwischen der Qualität der Bilderzeugung und der erreichten Robustheit. Wir zeigen, dass die gelernten Repräsentationen in den vorgeschlagenen regularisierten deterministischen Autoencodern mit einem vergleichsweise

billigen adversarischen Lernschema eine überlegene Robustheit gegenüber adversarischen Angriffen aufweisen, ohne die Qualität der Bilderzeugung zu beeinträchtigen.

Acknowledgements

First and foremost, I would like to thank my doctoral advisor, Prof. Margret Keuper, for her guidance and supervision during my research. Your advice and expertise have been instrumental in shaping the direction and scope of my research. Your insightful feedback and challenging questions helped me refine my ideas and pushed me to strive for excellence. I am very grateful for the time and effort you have invested during this time.

I would like to thank Prof. Michael Möller for reviewing my work and agreeing to be the co-examiner of this thesis. I would also like to thank the other members of the thesis committee for spending their valuable time.

I would like to give special thanks to my supervisor, Dr. Kathrin Skubch, who guided me throughout my research journey and for having confidence in me. I am grateful for all your insights, suggestions, and feedback during this time. Your constant encouragement helped me stay focused and motivated. I would not have been able to complete this dissertation without your support.

I also thank Dr. Stefan Falkner for his guidance, for proofreading the publications, and for providing valuable feedback. I enjoyed all our discussions, and they contributed significantly to improving the quality of the work. I extend my thanks to my colleagues at the Bosch Center for Artificial Intelligence where I spent the three years of my PhD. I am particularly grateful to Dr. Julia Vinodgraska for her constant care and support during my work at Bosch. I would also like to thank Dr. Christian Daniel for all the advice and help during my initial days at Bosch.

On a personal note, I am extremely grateful to my parents and sister for their love, encouragement, and constant support throughout this challenging journey. Thank you for motivating me when I was overwhelmed with publication deadlines or when things did not go as expected. I would also like to take this opportunity to thank my late grandfather, who inspired me to always dream big. Finally, I dedicate this thesis to my loving husband, best friend, and most enthusiastic cheerleader, Emil. Thank you for being patient and supportive even when I was sometimes unreasonable. Your belief in me and my abilities kept me going when I doubted myself. Your contribution was invaluable, and I couldn't have achieved this milestone without you.

Table of contents

List of figures

List of tables

List of symbols

The following table gives an overview of the symbols used in this thesis.

$\mathbb{R}$	the space of real numbers
$\approx$	is approximately equal to
$\leq$	is less than or equal to
$\in$	is member of
$\sim$	is distributed according to
sup	supremum
α	coupling parameter
Tr	trace of the matrix
x^T	transposed vector x
$\mathbb{1}$	indicator function
z	latent vector
z^{adv}	adversary latent vector
$\|\| \, . \, \|\|$	the absolute value
$\|\| \, . \, \|_2$	the l_2 norm of the vector
$\|\| \, . \, \|_\infty$	the l_∞ norm of the vector
$\mathbb{E}_x[\, . \,]$	expectation with respect to x
$\mathbb{N}(\mu, \sigma)$	multivariate normal distribution with mean μ and standard deviation *sigma*
σ	standard deviation of the Gaussian normal distribution
μ	mean of the Gaussian normal distribution
Σ	covariance matrix of the Gaussian normal distribution
Σ^{GMM}	covariance matrix of Gaussian Mixture Model prior
$\bar{\Sigma}$	empirical covariance matrix
$\bar{\Sigma}_{adv}$	empirical covariance matrix of adversaries
$\bar{\Sigma}^{cross}$	empirical cross-covariance matrix
$\int_a^b f(x)dx$	integral of a function with lower bound and upper bound b
$\prod$	product operator
$\sum$	summation operator
$\mathcal{L}(\, . \,)$	loss function

Chapter 1

Introduction

The ultimate goal of artificial intelligence (AI) is to understand the world around us and to enable machines to imitate human thought processes such as learning, reasoning, predicting, etc. This can only be achieved by identifying and understanding the underlying explanatory factors hidden in the available data. Deep generative models have revolutionized the field of AI by enabling computers to gain a deeper understanding of real-world data. These models have shown great potential in generating new data patterns that are almost indistinguishable by humans. Generative models have demonstrated exemplary performance in diverse applications, including computer vision [149, 82, 102, 151, 100, 99, 157], audio processing [109, 159, 186, 190], reinforcement learning [177, 117, 90], natural language processing [147, 148, 19, 138] and life science [123, 201, 60].

The process of creating new data has always been of particular interest in research, both because of the possibility of seemingly endless streams of new data and the implications of the knowledge that the model gains about the data manifold. The quality of the samples generated by deep generative models has improved tremendously in recent years. However, it is still not completely clear how exactly these models learn from the data, i.e., how well it encodes its features, biases, and properties that are meaningful to humans in the learned latent space [194]. Therefore analyzing and improving the latent representations of deep generative models still remains an active area of research [194, 99, 23, 22, 60].

1.1 Optimizing the Latent Representations

Latent representations are used to transform complex forms of the raw data surrounding us into simpler and more compact representations that are more convenient to process and analyze. The ability to learn good representations is a fundamental problem in machine learning to facilitate data-efficient learning. These representations, when properly learned,

can be used for a variety of downstream applications. It is also imperative that the learned representations should contain priors about the world around us, i.e., priors that are not task-specific but are likely to be useful for solving AI tasks. This is also the core concept of representation learning, where the goal is to discover interpretable latent representations [13]. The importance of representation learning in various domains has led to a growing research interest in learning latent representations with desirable features [23, 22]. Among the various ways of learning representations, we focus on latent representations of deep generative models in this thesis.

The effectiveness of generative techniques is highly dependent on the learned latent space or representations of the model. Desirable properties of the latent space include generalizability, controllability, smoothness, compactness, robustness, and disentanglement. For deep generative models, learning generalized, controllable, and disentangled representations of complex data distributions, such as images, helps in generating diverse, high-quality samples. Recently, it has also been found that a well-structured and smooth latent space of generative models enables efficient optimization of expensive black-box problems such as drug discovery, material design, and topology optimization [60, 140]. Optimizing the latent space of such models is therefore crucial to improve the quality and diversity of the generated samples and to further enhance the usability of the learned features for potential downstream applications.

This thesis aims to optimize the learned latent representations of deep generative models. We particularly focus on two popular classes of deep generative models, (i) Generative Adversarial Networks (GANs) [67] and (ii) Variational Autoencoders (VAEs) [107].

1.2 Contributions

A high-level overview of the research challenges addressed in this thesis and our contributions are given in Figure 1.1. The major contributions are listed below,

- We introduce a state-of-the-art adversarial network to enhance the controllability of GANs and generate high-quality images based on the numerosity of multiple objects present in the images [163].
- We propose a novel multi-modal regularization scheme to train simple and end-to-end trainable deterministic autoencoders as a potential alternative to VAEs to efficiently model complex data distribution [161].

- We introduce an effective and less expensive adversarial training scheme to the proposed deterministic autoencoders to enhance the accuracy and robustness of the learned latent space [162].

Each of these contributions is discussed in detail in the following section. This is organized into two sections; the first section deals with improving the controllability of GANs for content-based image generation applications, and the second section talks about optimizing the learned representations in VAEs.

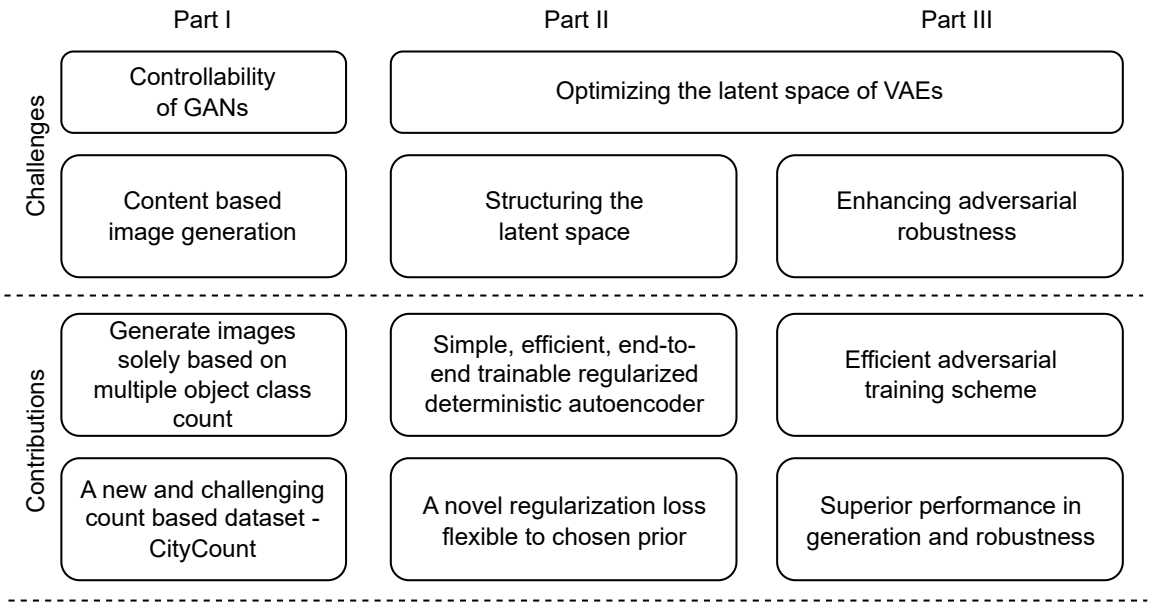


Fig. 1.1 A high-level overview of the research challenges addressed in this thesis, along with our contributions. In part I, we focus on improving the controllability of Generative Adversarial Networks (GANs) [163]. In parts II and III, we focus on optimizing the learned representations of Variational Autoencoders (VAEs). Part II introduces a novel regularized deterministic autoencoder as a promising alternative to VAEs [161]. In part III, we introduce methods to enhance the adversarial robustness of the proposed model [162].

1.2.1 Enhancing the Controllability of Generative Adversarial Networks

Since their introduction, Generative Adversarial Networks (GANs) have achieved remarkable feats in generating realistic images. With recent advances in the field, it is now possible to generate high-resolution, realistic images that are indistinguishable from real images [101, 102, 99, 18]. Considerable effort has also been made to control the content of the generated

Fig. 1.2 (Row 1) Real samples from our newly introduced real-world count-based CityCount dataset. (Row 2) Generated CityCount images by our model based on the multiple-class count input of cars and persons.

images. In the first part of the thesis, we take one step towards improving the controllability of the learned representations of GANs. We attempt to control the image generation process solely by conditioning the number of objects of predefined classes in the images, while a reasonable spatial layout is to be inferred from the training data distribution. Instead of addressing single object class counting as seen in [167, 184], where convolutional networks or recurrent neural networks are used to count, our approach focuses on counting object instances from multiple classes during *generation* as shown in Figure1.2. We introduce an extension to the StyleGAN2 architecture by incorporating an additional regression network into the discriminator to facilitate image generation based on the number of objects per class. Based on the findings in [30], our network uses dense blocks in the generator architecture to facilitate the propagation of the count constraint as well as the regression loss of the count network. By performing extensive experimental analysis on various datasets, including the proposed CityCount dataset, we show that our model can generate images with high fidelity based on the count constraint of multiple object classes without requiring expensive bounding box annotations of the objects. This work is in accordance with the ICCV, 2021 [163] publication entitled "Multi-Class Multi-Instance Count Conditioned Adversarial Image Generation" and is jointly supervised by Margret Keuper (University of Siegen) and Kathrin Skubch (Bosch Center for Artificial Intelligence).

1.2.2 Optimizing the Latent Space of Variational Autoencoders

In the second part, we examine the learned latent representations of Variational Autoencoders (VAEs). Although VAEs provide a strong theoretically backed framework for generative modeling, the practical constraints associated with its training and formulation limit the usability of these models. In part II and part III, we address the limitations of VAEs and introduce a deterministic autoencoder as a promising alternative to VAEs.

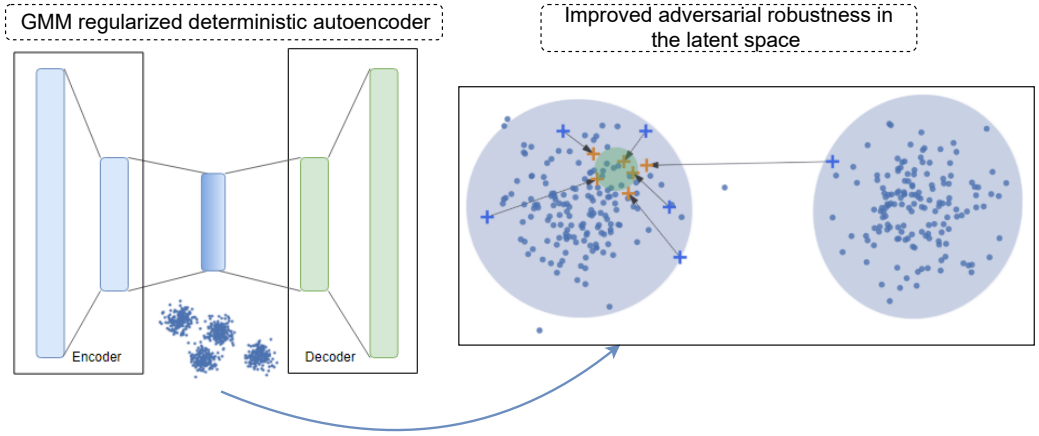


Fig. 1.3 Optimizing the latent space of Variational Autoencoders - (left) The proposed multi-modal regularization scheme efficiently structures the latent space of the deterministic autoencoder. (right) For ease of visualization, we consider two components of GMM prior and focus on the green shaded region. In order to improve the robustness of the learned latent space, we introduce an effective and inexpensive adversarial training scheme. During training, we enhance strong coupling between the adversarial samples (blue crosses) and their corresponding original samples (blue dots). During training, the adversarial samples move closer to the clean samples (orange crosses).

Structuring the latent space via regularized deterministic autonecoders Motivated by recent advances in deterministic autoencoders[60], our approach elegantly combines the idea of new training objectives with the extension to multimodal priors without increasing training complexity or compromising sampling quality, see Figure 1.3, left. We derive a strong training signal that can be derived in closed form for multimodal priors. In particular, we derive a novel deterministic regularization scheme from a strong metric for probability distributions. This ensures stable training and reliable regularization of the latent space and improves the quality of the samples. Through extensive empirical analysis, the proposed method could potentially learn better representations for applications such as image generation, drug molecule generation, and unsupervised clustering. This work corresponds to the

NeurIPS 2021 publication [161] titled "Shape your Space: A Gaussian Mixture Regularization Approach to Deterministic Autoencoders" and is jointly supervised by Margret Keuper (University of Siegen) and Kathrin Skubch (Bosch Center for Artificial Intelligence).

Enhancing the adversarial robustness of latent space Motivated by the promising potential of the proposed deterministic autoencoders in modeling both continuous and discrete data structures, we further investigate the robustness of the latent space of these models. To generate adversarial samples, we adapt the fast FGSM [196] method from the classifier literature to the latent space of the model. We introduce an adversarial training procedure (see Figure 1.3, right) to efficiently couple adversarial and original samples. The proposed learning scheme is comparatively less expensive and easier to implement than existing adversarially trained robust VAE models. We show that the deterministic formulation improves the robustness of VAEs against adversarial attacks when the latent codes are properly regularized. Unlike the existing robust VAE models, the proposed method ensures the robustness and fidelity of the learned representations. This work is in accordance with the NeurIPS 2022 publication [162] titled "Trading off Image Quality for Robustness is not Necessary with Regularized Deterministic Autoencoders" and is jointly supervised by Margret Keuper (University of Siegen) and Kathrin Skubch (Bosch Center for Artificial Intelligence).

1.3 Thesis Outline

This thesis is divided into six chapters. In the second chapter, we provide the preliminary information necessary for the following chapters. We begin this chapter with detailed insights into deep generative models, focusing on GANs and VAEs. This chapter also discuss the standard datasets and evaluation metrics that is utilized in the empirical evaluation of the proposed methods in the following chapters. The rest of the thesis is divided into two main sections and three parts, as shown in Figure 1.1. The first section corresponds to Chapter 3, and the second section to Chapters 4 and 5. In Chapter 3 (Part I), we focus on GANs; to be specific, we take a step towards improving the controllability of GANs for content-based image generation applications. Chapters 4 and 5 are devoted to improving the learned representations of VAEs. In Chapter 4 (Part II), we introduce regularized deterministic autoencoders as a possible alternative to stochastic VAEs. Motivated by the success of the model proposed in Chapter 4 in efficiently modeling complex data structures, we study the adversarial robustness of the learned latent space of the proposed model in Chapter 5

(Part III). Finally, we conclude the thesis in Chapter 6 by discussing the main findings and contributions of the proposed methods and potential future work.

1.4 Publications

The content of this thesis is based on the following publications.

Peer-reviewed conference papers.

A. Saseendran, K. Skubch, and M. Keuper. Multi-class multi-instance count conditioned adversarial image generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6742–6751, 2021.

A. Saseendran, K. Skubch, S. Falkner, and M. Keuper. Shape your space: A gaussian mixture regularization approach to deterministic autoencoders. In *Advances in Neural Information Processing Systems*, 2021, volume 34, pages 7319–7332, 2021.

A. Saseendran, K. Skubch, S. Falkner, and M. Keuper. Trading off image quality for robustness is not necessary with regularized deterministic autoencoders. In *Advances in Neural Information Processing Systems*, 2022.

Patents.

A. Saseendran, K. Skubch, and M. Keuper. Generator networks for generating images with predetermined counts of objects, US Patent App. 17/445,440. (Applied)

A. Saseendran, K. Skubch, S. Falkner, and M. Keuper. Shape your Space: A Gaussian Mixture Regularization Approach to Deterministic Autoencoders, US Patent App. 17/943890. (Applied)

A. Saseendran, K. Skubch and M. Keuper. A method for training deterministic autoencoders, EU Patent App. 22192386.5. (Applied)

1.5 Software

We provide an open-source implementation of the proposed methods in this thesis to promote open research and ensure reproducibility.

Count based image generation (Chapter 3 [163])
https://github.com/boschresearch/MCCGAN

GMM regularization based deterministic autoencoder (Chapter 4 [161])
https://github.com/boschresearch/GMM_DAE

Robust deterministic autoencoder (Chapter 5 [162])
https://github.com/boschresearch/Robust_GMM_DAE

Chapter 2

Preliminaries

2.1 Deep Generative Models

Generative modeling involves learning the probability distribution of a data manifold based on a representative sample set. A deep generative model captures the hidden patterns and structures in the data distribution to learn compact representations and use this knowledge to create new data. The compact, low-dimensional space that the model learns to represent the training data distribution is called the latent space. The latent variables in this space are often called the "hidden code" or "latent representation" of the data. The learned representations can then be used for various downstream applications such as image or video synthesis, text generation, music composition, data augmentation, anomaly detection, and medical imaging, to name a few. The existing pool of generative techniques includes Auto-Regressive [187, 29], Flow-based [154, 195, 82], Energy-based [47, 83], and Latent-Variable models [67, 107]. Among them, we are particularly interested in two popular Latent-Variable based models, Generative Adversarial Networks (GANs) [67] and Variational Autoencoders (VAEs [107]).

In Latent-Variable based models, the actual data distribution $p(x)$ is expressed through the marginalization over a vector z of latent variables as follows,

$$p(x) = \int_z p(x|z)p(z)dz = \mathbb{E}_{p(z)}[p(x|z)] \tag{2.1}$$

where z is the latent representation of a data point x distributed with a known distribution, also called prior distribution $p(z)$. The distribution, $p(x|z)$, is parametrized by a deep neural network. After training, the learned model is used to generate new samples via ancestral sampling, i.e., we first sample from the prior distribution, $z \sim p(z)$, and then generate new samples, $x \sim p(x|z)$.

This dissertation focuses on optimizing the latent representations of two popular deep generative models, Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs). The following section summarizes the concepts of these two models and some recent advances in generative modeling.

2.1.1 Generative Adversarial Network (GANs)

Generative Adversarial Networks, or GANs, are among the most popular generative models. The concept of GANs was introduced by Ian Goodfellow et al. in 2014 [67]. GANs consist of two network components, a generator and a discriminator. The goal of the model is to generate realistic samples that resemble the distribution of input data by training these two networks to compete against each other in a game-like manner.

Discriminator The discriminator is basically a supervised classifier that attempts to classify its input as either 'real' (1) or 'fake' (0). As shown in Figure 2.1, the inputs to the discriminator are real samples from the dataset or fake samples from the generator. The discriminator outputs a probability value for each sample input, indicating how likely it is to be a real sample.

Generator The generator is a deep neural network that receives a random noise vector as input, as shown in Figure 2.1 and is trained with the goal of generating samples that resemble the real data distribution. During the training process, the generator takes feedback from the discriminator and updates its weights accordingly during backpropagation. The generator gradually generates samples that are indistinguishable from the real samples to fool the discriminator.

The generator and the discriminator form the whole structure of GAN, as shown in Figure 2.1. In the training process, the generator is trained to generate samples that can fool the discriminator into thinking they are real, while the discriminator is trained to correctly classify its input samples as real or generated. Thus, the training of GANs can be viewed as a two-player game between the generator and the discriminator. After training, the Generator network is used to generate new data samples

Working Principle of GANs The training algorithm of GANs is summarized in Algorithm 1. At each step of the training process, we take a batch of m training samples and a batch of m latent vectors drawn randomly from a Gaussian/Uniform prior. The generator function is denoted by G with parameters θ_G, and the discriminator function is denoted by

D with parameters θ_D. Then the objective function of the GAN $V(G,D)$ is defined as follows,

$$\min_G \max_D V(D,G) \;=\; \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] \;+\; \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{2.2}$$

where $\mathbb{E}$ is the Expectation or expected value, and the distribution of data in the dataset (also called real distribution) is denoted by x.

During training, there are two simultaneous gradient steps, one to update the discriminator parameters θ_d to reduce the discriminator cost function J_D, and the other to update the generator parameters θ_g to reduce the generator cost function J_G. The goal of the discriminator is to maximize the probability assigned to real and fake samples. Thus, in mathematical terms, the discriminator cost function J_D is to maximize the average of the log-likelihood for the real samples and the log value of the inverted likelihoods for the fake samples, as given in Equation 2.2. The generator aims to learn a distribution P_g over data x. There are two different ways to formulate the generator's objective function. One corresponds to the minimax loss function, and the other to the non-saturating loss function. In the minimax objective, minimization corresponds to minimizing the generator loss, and maximization corresponds to maximizing the discriminator loss, as given in Equation 2.2. In this case, the generator seeks to minimize the logarithm of the inverse probability of the discriminator for fake samples, i.e., $J_G = \text{minimize}\log(1-D(G(z)))$. Intuitively, this means that the generator is encouraged to produce samples with a low probability of being fake. However, in the early stages of training, when the generated samples are not yet accurate or close to the real data distribution, the discriminator can distinguish between real and fake samples with high confidence. The discriminator wins easily, and the game ends. The model cannot be trained effectively in such a scenario. A non-saturating GAN loss was proposed to avoid saturation of the generator loss function. By slightly modifying the goal of the generator, i.e., instead of training G to minimize $\log(1-D(G(z)))$ we train G to maximize $\log(D(G(z))$. In contrast to the previous formulation, the generator now tries to maximize the probability that the generated samples are predicted to be real. This is computationally less expensive and also yields better gradients during learning. The number of steps k to train the discriminator is a hyper-parameter, and in the original work, it is suggested to use the least expensive option, $k = 1$ [67]. The choice of loss function for GANs is an active area of research. The most popular loss functions used in many implementations are least squares loss and Wasserstein loss[178].

The solution of the two-player game is a Nash equilibrium, which is the optimal point for the mini-max function of GANs [67]. The Nash equilibrium in game theory literature is the state in which no player can improve its individual gain by choosing a different strategy. When the discriminator receives a fake output from the generator $G(z)$, it tries to make

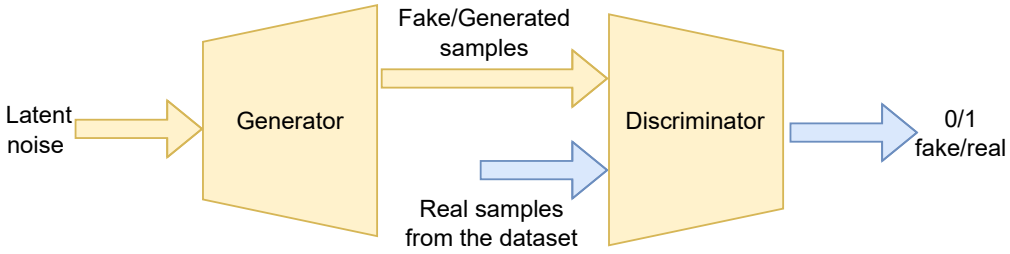


Fig. 2.1 GAN architecture comprising of the generator and discriminator. The generator takes in a noise vector randomly drawn from a chosen prior. The output from the generator and real samples from the dataset are given to the discriminator to classify as real or fake.

$D(G(z))$ equal to 0, while the generator tries to make it equal to 1. The Nash equilibrium would then be $G(z)$ drawn from the same distribution as the training data set, and $D(x) = \frac{1}{2}$ for all x. GANs are notoriously difficult to train because it is difficult to find an equilibrium and requires an exhaustive hyperparameter and architecture search. In the following section, we discuss some major constraints in training GANs.

One of the most common problems when training GANs is the so-called mode collapse, or sometimes Helvetica scenario, in which the generator collapses and produces only a limited variety of samples. When the generator finds that a particular mode or data type could easily fool the discriminator, it may start generating the same data type. There is no specific function in the generator's objective function that explicitly forces diversification of the generated patterns. If the generator repeatedly generates the same data type, it is best for the discriminator to learn to reject that particular output. However, suppose that in the next iteration, the discriminator gets stuck in a local minimum and does not find the best strategy. In this case, it is too easy for the next generator iteration to find the most plausible output for the current discriminator. Each generator iteration is over-optimized for a particular discriminator, and the discriminator never manages to learn from the trap. As a result, the generators rotate through a small set of output types. In practice, however, a complete mode breakdown is not expected. In contrast, partial mode collapse, where the generator outputs fewer different samples or fails to produce certain modes of data distribution, is a common phenomenon. Several techniques have been proposed to avoid mode collapse, such as using other objective functions like Wasserstein GAN or adding regularization terms to the generator or discriminator loss [71, 7].

The hyperparameters of the GANs must be chosen appropriately because there is a high probability that one of the networks will diverge or stop learning during training. Since training involves both networks, it is often observed that one of the networks is stronger than the other, which means that the gradient of the loss function can easily be zero. This is called

the vanishing gradient problem. Since the network depends on the hyperparameters, failure to determine the optimal values for these parameters can lead to an imbalance between the generator and the discriminator and, thus, overfitting. Several methods and architectural changes have been proposed to stabilize the training of GANs, some of which are discussed in the next section. Among the many potential applications that use GANs, we mainly focus on image generation applications, which are discussed in detail in the next section.

Algorithm 1 GAN algorithm

```
for number of training iterations do
    for k steps do
        Sample batch of m noise samples from noise prior p_g(z).
        Generate m samples from the noise prior.
        Sample batch of m samples from the training dataset.
        Update discriminator parameters.
    end for
    Sample batch of m noise samples from noise prior p_g(z).
    Generate m samples from the noise prior.
    Update generator parameters.
end for
```

Image generation using GANs Since their introduction, GANs have rapidly evolved to become the most promising trend for generating diverse photo-realistic images. Deep convolutional GAN (DCGAN) [146] demonstrated the potential of convolutional neural networks in this context for the first time. Conditional GANs(cGANs) [133] extend the Vanilla GAN architecture by conditioning the generator and discriminator on additional information, such as class labels or other attributes. This allows the model to generate samples that belong to specific classes or have certain attributes. Arjovsky et al. [71] introduced the Wasserstein GAN (WGAN), which uses the Wasserstein distance to measure the difference between the generated and real data distributions. WGANs have been shown to be more stable and easier to train than traditional GANs, as they avoid the problem of mode collapse and vanishing gradients. CycleGANs [209] are used for image-to-image translation applications, where the goal is to translate an image from one domain to another (e.g., turning a summer landscape into a winter landscape). CycleGANs consist of two GANs, each with a generator and a discriminator, that are trained to translate images in both directions and are trained with an additional cycle consistency loss function to ensure that the translated images are consistent with the original images. A considerable amount of research was also devoted to improving the training stability of GANs [71, 99, 134] and to develop more evolved architectures [18, 101, 102, 146]. Progressive GANs(PGANs) [99]

use a progressive training procedure that starts with a low-resolution generator and gradually increases the resolution as training progresses. This allows the model to generate high-quality images with fine details. In 2018, researchers at Google introduced BigGAN, a variant of GAN designed to generate high-quality images with a resolution of 256x256 or 512x512 pixels. BigGAN [18] uses a hierarchical generator and a novel truncation trick to improve the quality and diversity of generated images. StyleGAN [101] extends the progressive growing architecture [99] for both the generator and discriminator to generate high-resolution images such as of 1024×1024 resolution. These advancements in GAN research have led to significant improvements in the quality and diversity of generated outputs, training stability, and increased control over the generated outputs.

2.1.2 Variational Autoencoders (VAEs)

Autoencoders Autoencoders [164, 9] are a class of neural networks trained to reconstruct the input data while learning a compact, low-dimensional representation of the input. The model consists of two components, the encoder and the decoder. The goal of the encoder is to learn a latent representation of the input data such that the variational factors in the data are captured so that the decoder can reconstruct them. When learning the latent space in an autoencoder, there are no specific constraints as long as the model can reconstruct the input when the decoder function is applied. Autoencoders are commonly used in applications such as data compression, dimensionality reduction, anomaly detection, and image denoising [31, 3, 64, 55]. Although autoencoders can learn powerful representations of the input data, they are not suitable for generating new data samples due to the non-regularized latent space.

Variational Autoencoders (VAEs) Variational Autoencoders (VAEs) have a structure similar to that of classical Autoencoders. However, VAEs impart generative capabilities to the latent space by learning the underlying training data distribution. The key idea behind VAEs is to learn a probabilistic mapping of the data space to a latent space and then use this mapping to generate new samples that resemble the original data.

The VAE framework consists of two network components: the encoder and the decoder, as shown in Figure 2.2. The encoder maps the input data sample to a probability distribution in latent space. The output of the encoder is the parameters of the latent space distribution, the mean, and the variance. The decoder samples from the latent space distribution and provides an output similar to the input data. After training, the decoder is used to generate new samples. Encoder and decoder models are parameterized by deep neural networks.

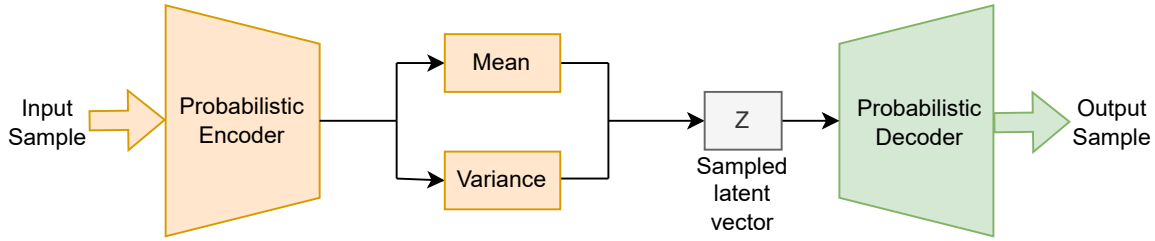


Fig. 2.2 VAE architecture comprising of the encoder and decoder.

VAE Formulation Given a set of training data $X = \{x^1, x^2, ..x^N\}$, the task of the generative model is to generate new data samples $x \in \mathbb{R}^d$. The main purpose of VAEs is to learn a low-dimensional representation of the input data points, often referred to as latent variables, denoted by $z \in \mathbb{R}^k, k < d$. We then consider a latent-based model (stochastic decoder) $p_\theta(x|z)$ with prior $p_\theta(z)$, where θ corresponds to the parameters of the decoder model. The prior distribution is assumed to be the standard normal distribution. The objective is to maximize the probability of each data in the training dataset X, which is defined as follows,

$$p_\theta(x) = \int p_\theta(x,z)dz = \int p_\theta(x|z)p_\theta(z)dz \tag{2.3}$$

However, the integration in Equation 2.3 is performed overall dimensions of z and is intractable. Therefore, for VAEs, the distribution p_z is derived using the posterior $p(z|x)$, which is inferred using a variational inference approach [16]. We first model $p(z|x)$ with another, simpler and easy to find distribution $q(z|x)$. This is achieved by minimizing the divergence between these distributions,

$$q_\phi(z|x) = \arg\min_q D_{KL}(q_\phi(z|x) \parallel p_\theta(z|x)) \tag{2.4}$$

where $q_\phi(z|x)$ is defined as the stochastic encoder or inference model with parameters ϕ that approximates $p_\theta(z|x)$, and D_{KL} corresponds to the Kullback-Leibler (KL) divergence measure. From the definition of KL divergence and by rearranging the terms in Equation 2.4, the final objective of VAEs is derived as follows,

$$\ln p_\theta(x) = -D_{KL}(q_\phi(z|x) \parallel p_\theta(z)) + \mathbb{E}_{q_\phi(z|x)}[\ln p_\theta(x|z)] \tag{2.5}$$

Equation 2.5 defines an alternative definition for $p_\theta(x)$ that does not require knowledge of $p_\theta(z|x)$. The first term in Equation 2.5 corresponds to the regularization loss in the latent space. The second term corresponds to the data reconstruction loss, which measures the difference between the input and reconstructed data of the decoder. With a Gaussian

prior assumption, the reconstruction loss corresponds to the maximization of the Gaussian likelihood, which corresponds to the squared distance between the input and output data samples. The regularization loss is defined as the KL divergence between the encoder output and the prior distribution, which is non-negative and is zero when the encoder distribution matches the prior distribution.

The right-hand term in Equation 2.5 is called the evidence lower bound (ELBO) [107]. To optimize this bound with respect to the parameters θ and ϕ, gradients must be back-propagated through the stochastic sampling process of the latent variable z. This is made possible by introducing a reparametrization technique [107] in which the latent variable z is reparametrized so that the stochasticity is independent of the parameters of the distribution. This is achieved by introducing an auxiliary noise variable $\varepsilon \sim \mathcal{N}(0,1)$ and redefining z as $z = \mu(x,\phi) + \sigma(x,\phi)\varepsilon$.

Despite the strong theoretical formulation, VAEs tend to produce unrealistic images and blurry reconstructions when applied to complex image datasets [17, 46]. This is attributed to the maximum likelihood objective function and Mean Squared Error (MSE) reconstruction loss in the optimization strategy. There is also evidence that the limited approximation to the true posterior is the cause of this problem [208, 185] with the MSE term strongly favoring non-Gaussian posteriors. Some other limitations associated with VAEs include over-regularization due to the KL divergence term in the objective function and simplified prior assumptions during training [60, 17].

VAE variants Since the introduction of VAEs, many follow-up works have tried to overcome the practical and theoretical limitations of the framework, e.g. [14, 178, 179], and make them applicable to specific applications such as image generation [26, 193, 141, 151, 60, 155, 172, 207, 124], clustering [39, 144] or anomaly detection [211]. Higgins et.al [79] improved the learned representation in VAEs by encouraging disentanglement in the latent space and introducing β-VAE for disentangled factor learning. β-VAE modifies VAEs by introducing a hyperparameter to balance the latent regularization term with the reconstruction performance. Chen et.al [26] further decompose the ELBO term in VAEs to introduce total correlation (TC) regularization and propose β-TCVAE as a promising alternative to β-VAEs. Willets et al. [193] show that adding the TC term to the VAE objective also improves the robustness of the learned representations in VAEs.

In the standard VAE framework, the prior distribution is commonly assumed to be a Gaussian normal distribution. This might lead to simplified representations learned by the model, which cannot represent the rich semantics in the data distribution. Several methods were also proposed to include complex and flexible priors to the training pipeline of

VAEs to enhance the semantics of the learned latent representations [28, 211, 179]. Miao et al. [131] introduce an approach to incorporate the inductive bias into VAEs without explicitly changing the prior by utilizing an intermediary set of latent variables. Hierarchical VAEs [155, 172, 207, 124] extend the standard VAE framework by introducing a hierarchy of latent variables and offering superior modeling capabilities. Casale et al. [21] employs Gaussian process priors to account for correlations between the data samples. In [70], a Bayesian non-parametric prior is used with a hierarchical non-parametric variational autoencoder for video representation learning. Chen et al. [28] use an auto-regressive prior to achieving improved generative performance on image datasets. Berger et al. [14] propose to replace the standard spherical Gaussian prior with a more general version with an arbitrary covariance matrix and learn the correlations by optimizing the evidence lower bound of the model.

In another line of work, multi-modal priors were utilized in VAE models. Zong et al. [211] propose to use a GMM prior in autoencoders for unsupervised anomaly detection by training an additional network estimating the parameters of the GMM. Lee et al. [116] address unsupervised meta-learning using a GMM prior in VAEs to shape the latent space by employing an extension of the evidence lower bound to complex variational inference schemes. Tomczak et al. [179] propose to replace the GMM prior by coupling the posterior and prior of the model. Adversarial autoencoders [127] improve the generative performance of VAEs by incorporating adversarial learning into the VAE framework and offer competitive performance in image generation at an increased computational complexity and decreased training stability. To account for the over-regularization effect of the KL divergence term in the standard VAE framework, [178] minimize the Wasserstein distance between the representations learned by the model and the target prior. The state-of-the-art VAE model for high-fidelity image generation, VQ-VAE [141, 151], involves two stages of training relying on complex discrete autoregressive density estimators. Gosh et al. [60] question the variational formulation of VAEs and introduce a simple and effective deterministic model without any prior assumptions, followed by a post-hoc density estimation to approximate the learned posterior. The authors use the negative log-likelihood for regularization but require a post-hoc step to derive a strong sampling procedure from the model. More details on these models are provided in the next section.

Regularized Autoencoders (RAEs) Regularized autoencoders [60](RAEs) question the variational framework adopted by the VAEs and propose a deterministic approach to achieve comparable or better image generation performance than VAE-based models. Although VAE presents a theoretically sound framework for modeling the input data, some of the

approximations associated with the variational framework pose practical challenges during training. One of the significant drawbacks observed is the unsatisfying compromise between the quality of the generated and reconstructed samples. Prior posterior mismatch and one-sample approximation are two major limitations associated with the low sampling quality in VAEs.

- Prior-Posterior mismatch - In VAEs, the prior distribution over the latent variables is often chosen to be a simple distribution such as a standard normal distribution. However, given the data, the true posterior distribution over the latent variables may be much more complex. This mismatch between the prior and the true posterior is known as the prior-posterior mismatch problem. This mismatch can lead to poor sampling performance of the VAE.
- One sample approximation - While training VAEs, since the actual posterior distribution over the latent variables is intractable, an approximate posterior distribution is used instead. In theory, many samples must be drawn from the posterior to approximate the distribution. However, in practice, a one-sample approximation is carried out, which involves sampling one point from the approximate posterior distribution and then decoding that sample to generate a sample from the data distribution. This approximation leads to slow learning and sampling quality issues in VAEs.

Various techniques have been proposed to address these issues, such as using more flexible prior distributions or adjusting the architecture of the VAE. RAEs take a different approach to these methods and redefine a deterministic autoencoder as a generative model. VAEs can be considered deterministic autoencoders with noise injected into the decoder. Hence a deterministic encoder-decoder pair is trained with a regularization scheme instead of this noise injection to obtain a smooth latent space. The training objective of RAEs is to minimize the following loss function,

$$\mathcal{L}_{\text{RAE}} = \mathcal{L}_{\text{REC}} + \beta \frac{1}{2} \|\mathbf{z}\|_2^2 + \lambda \mathcal{L}_{\text{REG}} \tag{2.6}$$

where β and λ are the hyperparameters. The $\mathcal{L}_{\text{REC}}$ term corresponds to the reconstruction loss between the decoder and encoder, and $\mathcal{L}_{\text{REG}}$ is explicit decoder regularization such as L2-regularization or spectral normalization [134] used in GAN models. The second term in the loss function is used to constrain the size of the learned latent space to prevent unbounded optimization. Finally, to enable the generative mechanism of the model, an ex-post density estimation is performed on the learned latent representations after the training. A full covariance multivariate Gaussian with a 10-component Gaussian mixture model is used as the density estimator.

2.1.3 Recent Advancements

Research in generative modeling has grown exponentially in recent years. In this section, we briefly review some recent advances in the field of generative modeling. In particular, we focus on some of the most promising approaches that have attracted considerable attention in the computer vision community, such as flow-based models, transformer-based models, and diffusion models.

Flow-based models [154, 106] are generative models that use a sequence of invertible transformations to map a simple known base distribution (such as the normal distribution) to a more complex target distribution. Since their introduction, these models have been widely used in various application areas such as Computer Vision [106, 112, 104, 1], Natural Language Processing [183, 210, 94], and Reinforcement Learning [166, 129, 180]. Flow-based models allow exact likelihood computation using normalizing flows. The general idea is to map the unknown distribution in the input space to a known distribution in the latent space using an invertible function. The invertibility of the transformations allows for efficient inference and sampling, and the flexibility of the transformations allows for the modeling of complex distributions. The latent space of these models is not low-dimensional due to the constraints imposed by the invertible function and therefore requires high computational power and slow training time, especially for high-dimensional data. Despite recent developments in this area for image generation applications [41, 42, 106, 82, 195], the quality of sampling still suffers compared to the powerful GAN/VAE variants or the recently developed diffusion models.

Transformers are extremely effective in solving a variety of machine learning tasks and have been successfully applied to text sequences [188, 37, 147, 148, 19], images/videos [45, 51, 93, 75, 49, 168], speech [191], protein sequences [156], graphs [189], and time series [197]. Their greatest successes have been in building language models [147, 148, 19] and, more recently, in replacing convolutional networks in computer vision [49, 75]. Transformer models use self-attention mechanisms to capture dependencies between input features. Although previous work has used attention methods, transformer models are distinguished by the multi-head attention mechanism optimized for parallelization. Unlike convolutional neural networks (CNN), transformers have no inductive bias, which allows these models to learn long-range dependencies in the training distribution. Although transformer models were originally proposed for natural language processing (NLP) applications [188], the breakthrough of these models in the NLP domain has generated much interest in the computer vision community, especially in the area of generative modeling [49, 142, 25, 15, 93]. Motivated by the success of the GPT model in the NLP domain [147, 148, 19], iGPT [25] was proposed to extend the same architecture for image generation and achieved impressive per-

formance over the unsupervised CNN models. In [49], Esser et al. proposed to integrate the inductive bias of CNN and the expressive power of transformers to generate high-resolution images. Hybrid models such as TransGAN [93] with a transformer architecture for both the generator and discriminator of GAN have also been proposed to achieve comparable or better performance than CNN-based GAN counterparts. Recently, it has also been shown that transformer architectures are capable of generating high quality images for a given text description [150, 149]. Although transformer-based models have achieved impressive performance in various application domains, the major bottlenecks include the requirement for large amounts of training data and associated high computational costs [103].

Diffusion models are powerful probabilistic generative models that have displayed their exquisite potential in the field of computer vision [12, 61, 84, 83, 149, 150, 33], sequence modeling [118, 175], audio processing [109], and life science applications [123, 201]. These models define a nonlinear mapping from latent variables to the observed data where both quantities have the same dimension. Similar to VAEs, diffusion models approximate the data likelihood using a lower bound based on an encoder that maps the input to the latent variables. However, the encoder is predetermined, and the objective of these models is to learn a decoder which is the inverse of the process. The encoder or the forward diffusion process uses a sequence of diffusion steps to map the input through a series of intermediate latent variables. In this process, the data is gradually mixed with noise and repeated until only noise remains. The decoder or the reverse diffusion process learns the reverse process to map the data back through the latent variables, removing noise at each stage. New samples are generated by sampling noise vectors and passing them through the decoder. One of the main advantages of diffusion models is their ability to produce high-quality images with realistic textures and details [138, 38, 149]. Diffusion models have also been used in combination with other techniques, such as attention mechanisms [150, 149] and progressive training [57], to further improve their performance. Since the diffusion model operates in pixel space, they are limited by the computational cost of training and inference. Recently introduced latent diffusion models [157] aim to overcome the computational limitations of diffusion models by applying them in the latent space of powerful pre-trained latent spaces of autoencoders. Despite these advances, these models still require a large number of diffusion steps to produce high-quality samples, which can be computationally intensive for high-dimensional input spaces[33].

2.2 Commonly Used Dataset

This section summarizes the commonly used dataset and the evaluation metric used for the empirical analysis of the proposed methods.

2.2.1 MNIST

The MNIST (Modified National Institute of Standards and Technology database) [36] is a widely used dataset for computer vision and deep learning research, including generative models. The dataset consists of a training set of 60,000 28x28 grayscale images of handwritten digits (0-9) and a test set of 10,000 images along with corresponding digit labels; please refer to random samples of the dataset in Figure 2.3. The dataset includes 28×28 pixel grayscale images with 60000 training images and 10000 testing images. The MNIST dataset is considered a benchmark dataset in the field of machine learning, particularly in the area of image recognition. Many researchers use the MNIST dataset as a testbed for developing and evaluating new machine-learning models. It is small enough to be easily trained on most computers but complex enough to provide a challenging problem for machine learning algorithms.

The dataset was generated from another NIST database comprising binary images of handwritten digits collected from Census Bureau employees and high-school students. The black and white images from the NIST database were normalized and anti-aliased to generate a 28×28 grayscale image. Since the dataset contains label information, MNIST images are used in supervised learning tasks such as digit recognition or classification. Since the dataset is most commonly used for image-based tasks, it is a natural choice for evaluating the performance of generative models on image-generation tasks. Hence in this dissertation, we utilized MNIST images as an effective baseline to analyze the potential of the generative models and to conduct ablation studies to analyze the proposed methods.

2.2.2 FASHIONMNIST

The Fashion-MNIST [199] dataset, introduced by Zalando, is a dataset of images of clothing items, such as shirts, trousers, and bags. Each image is 28x28 pixels and is associated with a label indicating the type of clothing item it represents. The dataset is intended to replace the commonly used MNIST dataset, which consists of images of handwritten digits, and is often used as a benchmark for image classification tasks. The goal of using the Fashion-MNIST dataset instead of the MNIST is to provide a more challenging problem for machine learning models and a dataset more representative of real-world use cases. The dataset contains

60,000 training images and 10,000 test images and is split into 10 classes, each representing a different type of clothing item. The classes are T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle boot. Figure 2.3 shows random samples of the dataset. The images in this dataset are well-labeled and straightforward to process and hence widely employed in applications like image classification/detection and generative modeling.

Although the MNIST dataset is considered a good benchmark for machine learning or deep learning applications, there are some limitations associated with the same. Since the dataset is too simple, deep learning models could quickly achieve 99% accuracy on these images. Hence, the Fashion-MNIST dataset allows researchers to train and evaluate machine learning models for more challenging use cases.

2.2.3 SVHN

The Street House View Numbers (SVHN) [137] is a real-world dataset of house numbers represented by individual digits from 0 to 9. The dataset was created by collecting images from Google street view images and Amazon Mechanical Turk (AMT) framework to identify single digits in the images. The images are taken from various angles and under varying lighting conditions, which makes the dataset useful for training models for real-world computer vision tasks. It contains over 600,000 digit images, annotated with bounding boxes around the individual digits. The dataset is divided into three sets: a training set of over 500,000 images, a testing set of over 10,000 images, and a smaller set of additional images that can be used for additional training or validation. The dataset comes in two formats, (1) Original full house number images with varying resolution and (2) cropped images as in MNIST, centered around a single character with 32×32 pixels. The images might include overlapping digits and noisy or distracting features. Figure 2.3 shows random samples from the dataset. As seen in the figure, the images are similar to MNIST images with cropped digits; however, with more complexity and depict a much more complicated problem of recognizing digits or numbers from complex real-world images. SVHN also provides a large amount of labeled data, providing a reliable real-world benchmark dataset for deep learning applications. In this dissertation, we used the cropped version of SVHN images without the additional samples for evaluation.

2.2.4 CELEBA

The Large Celeb Faces Attributes (CelebA) [120] dataset is a large-scale collection of celebrity face images collected from the internet. The dataset includes more than $200,000$, 178×218 pixel size images of 10177 celebrities. Each image is annotated with 40 binary

Fig. 2.3 Random samples of images from each of the datasets considered for empirical evaluations in this thesis.

attributes such as "wearing glasses" or "has long hair" and five landmark points for the eyes, nose, and mouth. Figure 2.3 shows random samples from the dataset. Due to the large variety with different backgrounds, poses, and rich annotations, CELEBA images are widely used for various deep learning applications such as face detection/localization and generative modeling. In this dissertation, we center-cropped the original images and then resized them to 64×64 pixels for the image generation experiments.

2.2.5 Evaluation metric

Developing an appropriate metric to analyze and compare the performance of deep generative models is a challenging problem [17]. Qualitative analysis of generated samples is a widely used method in most related work, however, it is time-consuming and subjective to compare many works. Two of the most commonly used metrics for the quantitative evaluation of generative models are Inception Score (IS) [160] and Fréchet Inception Distance (FID) [78].

FID measures the distance between the feature vectors of the real and generated images to quantify their similarity. To be precise, the activations of the last fully connected layer, also called the global spatial pooling layer of an Inception V3 model, are extracted to compare real and generated images. The obtained activations are modeled as multivariate Gaussian distributions by computing the mean and covariance of the images. These activations or feature vectors are then computed for a collection of real and generated images, and the distance between them is called the Fréchet distance or Wasserstein-2 distance. Since the inception model computes this distance, the metric is called Fréchet Inception Distance. Lower values of FID indicate better quality, as they correspond to a small distance between the generated and real images. Conversely, a higher FID value indicates a lower quality or lower similarity between the generated images and the real images.

FID was originally proposed as a potential alternative to another evaluation metric called Inception Score (IS), which is used to determine the quality and diversity of the generated images. IS also uses the Inception V3 model to calculate the corresponding score. The IS score is calculated based on the classification performance of the Inception model for the synthetic images to assign the images to one of the 1000 known object classes. However, unlike FID, IS does not provide information about how similar the generated images are to the real data. Other alternative metrics, such as Perceptual Path Length, Kernel Inception distance, and Precision vs. Recall, are also proposed to evaluate the performance of deep generative models. FID, however, is one of the most widely accepted metrics in the literature to standardize the performance of generative models. Therefore, in this dissertation, we compute the FID score to quantitatively assess the quality of the generated images in our empirical evaluation.

FID Calculation A pre-trained Inception v3 extracts the feature vectors for both the real and generated images. The activations of the last fully connected layer of this model have a size of 2048, so the extracted feature vectors for real and generated images have the same dimension. The FID is then calculated by measuring the distance between these 2048 feature vectors. The feature vectors are modeled as multivariate Gaussian with mean, μ, and covariance, C. Let μ_r, C_r, and μ_f, C_f be the feature-wise mean and covariance matrix of the real and generated images; FID is then defined as follows,

$$d^2(\mu_r, C_r)(\mu_f, C_f) = \|\mu_r - \mu_f\|^2 + \mathrm{Tr}(C_r + C_f - 2\sqrt{C_r C_f}) \tag{2.7}$$

where Tr refers to the trace linear algebra operation (sum of the elements along the main diagonal) of the covariance matrix.

Chapter 3

Multi-Class Multi-Instance Count Conditioned Adversarial Image Generation

Image generation has rapidly evolved in recent years. Modern architectures for adversarial training allow the generation of even high-resolution images with remarkable quality. At the same time, more and more effort is dedicated to controlling the content of generated images. In this chapter, we take one further step in this direction and propose a conditional generative adversarial network (GAN) that generates images with a defined number of objects from given classes. This entails two fundamental abilities (1) being able to generate high-quality images given a complex constraint and (2) being able to count object instances per class in a given image. Our proposed model modularly extends the successful StyleGAN2 architecture with count-based conditioning and a regression sub-network to count the number of generated objects per class during training. In experiments on three different datasets, we show that the proposed model learns to generate images according to the given multiple-class count condition, even in complex backgrounds. In particular, we propose a new dataset, CityCount, derived from the Cityscapes street scenes dataset, to evaluate our approach in a challenging and practically relevant scenario. This work is published in the International Conference on Computer Vision (ICCV), 2021 [163].

3.1 Introduction

Developmental studies show that the human brain is endowed with a natural mechanism for understanding numerical quantities [35, 198]. Even young children have an abstract

understanding of numeracy and can generalize the concept of counting from one category to another (*e.g.* from objects to sounds) [198]. While counting object instances are relatively easy for humans; it is challenging for deep learning and computer vision algorithms, especially when objects from multiple classes, e.g., persons and cars, are considered. In this chapter, we take a step towards such elementary visual reasoning by addressing the generation of images conditioned on the number of object instances per object class. We are particularly interested in the complex case where objects from *multiple classes* are present in the same image. This is a fundamental vision task, which can even be solved by small children [35], but remains an unsolved problem in computer vision. Apart from that, many practical applications can benefit from the capability to generate images respecting numerical constraints. It especially aids the generation of additional diverse training data for visual question-answering and counting approaches. Further, the generation of technical designs based on the number of different components is of particular interest in the field of topology design, where data-based approaches have recently been explored successfully in applications ranging from molecular design [5] for chemical applications to product design [140] for aesthetics or engineering performance.

We propose to solve *multiple-class count* (MC^2) conditioned image generation (*i.e.* the generation of images conditioned on the number of objects of different classes that are visible in the image) as a modular extension to the state-of-the-art network for adversarial image generation, StyleGAN2 [102]. We further argue that object counting should be considered a multi-class regression problem. While this approach is simple, it allows the similarity between neighboring numbers to be naturally encoded in the network and to transfer the ability to count from one class to another. This will ideally make our network learn to generalize the concept of counting from one object class to another, meaning that it can see images of "two cars and one person" at training time and deduce the appearance of "two persons" at inference time. To the best of our knowledge, this is the first attempt to evaluate the potential of GANs to generate images based on the multiple object class count.

We validate the proposed approach in two lines of experiments. First, we evaluate the generative performance of our model on synthetic data generated according to the CLEVR [96] dataset as well as on real data from the MNIST [36] and SVHN [137] dataset. We propose a new, challenging real-world dataset, CityCount, derived from the well-known street scenes dataset Cityscapes [32]. The CityCount dataset comprises various crops from Cityscapes images containing specific objects from the important classes, *car* and *person*. The dataset includes various challenging scenarios such as diverse and complex backgrounds, object occlusions, varying object scales, and scene geometry. Samples from the CityCount dataset and generated samples from our model are shown in Figure 1.2. In the second line of

experiments, we show that the images generated by MC^2-StyleGAN2 can enhance the size and quality of training data for count prediction networks trained on images from CLEVR and CityCount.

3.2 Related Work

In this section, we start by discussing conditional GANs and some of the seminal works in this field. Since we focus on count-based image generation, we also review counting approaches developed for various computer vision applications.

3.2.1 Conditional GANs

Conditioning GANs (CGAN) on explicit information was first introduced by Mirza *et al.* [133]. Since then, various approaches have been proposed to improve the controllability of GANs. Many of these require extensive additional information such as class labels and/or natural language descriptions, e.g., image captions for text-to-image or text-to-video generation [8, 81, 133, 152]. Other variants of conditioning GANs include an information-theoretic extension to GANs (InfoGAN) [27], auxiliary classifier GAN (ACGAN) [139], twin auxiliary classifier GAN (TACGAN) [66] and projection based conditioning methods [135]. ACGAN extends the loss function of GAN with an auxiliary classifier to generate images. TACGAN further improves the divergence between real and generated data distribution of ACGAN by an additional network that interacts with the generator and discriminator. In projection-based methods [134], the condition is projected to the output of the discriminator by considering the inner product of the conditional variable and the feature vector of images. ContraGANs [97] introduces a conditional contrastive loss to learn the relation between input images. SpatialGAN [81] proposes a method for multiple conditioning with bounding box annotations and class labels of objects, and image captions to control the image layout in terms of object identity, size, position, and number. In their method, object bounding boxes are provided at test time, so the idea of count does not need to be learned. In [50], the authors propose a variational U-Net architecture to condition the image generation on shape or appearance. Various approaches have also been suggested to control the image generation process of GANs in applications such as image-to-image-translation [89, 209] or attribute transfer [76, 119]. Our work is related to ACGAN, with focus on the problem of multiple-class counting using regression.

Based on the high-resolution architecture introduced in [99], StyleGAN [101] employs adaptive instance normalization [86] based feature map re-weighting to facilitate the ma-

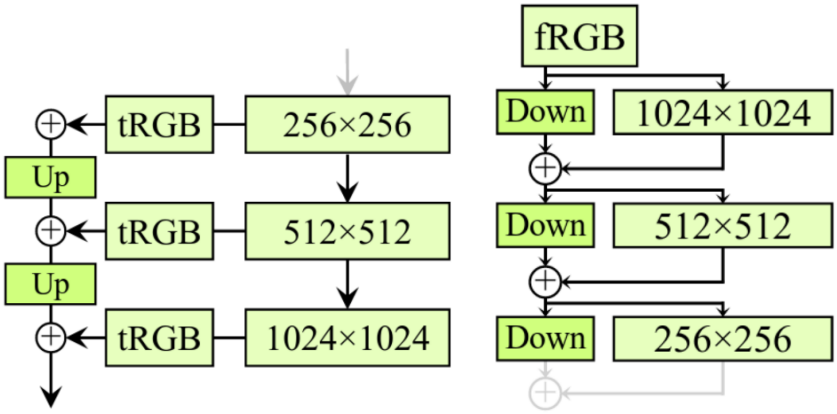
nipulation of images over multiple latent spaces, encoding different style properties. StyleGAN2 [102] improves over StyleGAN and avoids some characteristic generation artifacts. Recently, a new technique was proposed [100] to achieve state-of-the-art results with StyleGAN2 even when the training data is limited. While these approaches allow implicit conditioning of image contents, for example, on given styles, they do not enable to steer explicit properties of a generated image, such as the number of generated object instances per object class. Our proposed model introduces an extension to StyleGAN2 that facilitates such explicit conditioning.

3.2.2 Counting approaches

One way to count objects in an image is to localize and classify them using an object detection network and then count all found instances. While this approach is effective, it also requires additional class labeled bounding box or object prototype information [24, 170]. Adapting these approaches for conditional image generation will require additional information, such as pre-defined locations of the objects of interest during training. Other methods rely on recurrent neural network architectures and attention mechanisms [153, 158, 204]. Thus, they can not easily be applied in our problem setting. Density estimation-based counting methods [53] show that learning to count can be achieved without prior detection and are more reliable in severe occlusion scenarios. Multiple approaches have been proposed to counting object instances in images, for example, in the context of visual question answering [6, 111, 192]. In [2], Agarwal *et al.* suggests generating training data for this task by modifying the number of objects using cropping and inpainting. ARIGAN [62] utilizes a conditioned DCGAN to generate images of plants given the number of leaves.

3.2.3 StyleGAN - A Style-Based Generator Architecture for Generative Adversarial Networks

The StyleGAN [101] proposes a novel approach to modify the generator of GANs to enhance the controllability during image generation. The model yields state-of-the-art performance in unconditional image generation. The architecture of the model is based on the Progressive GAN (ProGAN) [99], which employs progressive training of the generator and discriminator starting with a low resolution (4×4) at the first layer and gradually increasing the resolution (e.g., 1024×1024) for high-resolution image synthesis. The intuition behind this architecture is to learn base or low-level features at the initial stages and gradually focus on complex-level features at higher resolution. These models are highly effective in learning superior-quality high-resolution images but are limited in performance when controlling the specific features



(a)Skip generator. (b)Resnet discriminator.

Fig. 3.1 StyleGAN2 Generator and Discriminator. tRGB and fRGB convert between RGB and high dimensional per pixel data. Up and Down corresponds to upsampling and downsampling operations. The diagram is taken from [102]

of the generated images. StyleGAN architecture was proposed to overcome this limitation of ProGAN models and to facilitate unsupervised separation of high-level attributes and stochastic variation in the generated images.

The StyleGAN generator is inspired by the style transfer literature, where neural representations are utilized to separate and recombine the style and content of the generated images. The input to the network is a learned non-linear mapping from the latent codes. The generator includes two different components, a *mapping network* to learn an intermediate latent space from the latent codes and a *synthesis network* where the learned style components are injected to multiple convolution layers via Adaptive Instance Normalization(AdaIN) [87]. In AdaIN, each feature map, x_f, is initially normalized by its mean and variance. The normalized feature maps are then scaled and biased by the corresponding scale and bias components of the style vector s at level i. Gaussian noise is added to each convolution layer to generate stochastic details in the images. Each convolution component in the synthesis network includes two AdaIN operations with two style injections and external noise additions resulting in multichannel images with resolution doubling at each block. The discriminator of StyleGAN also consists of a progressive growing architecture similar to that of ProGAN.

StyleGAN2 StyleGAN2 [102] is an upgraded version of StyleGAN with significant improvement in the quality of the generated images. StyleGAN2 revisits the generator normalization and progressive training to eliminate some artifacts observed in the StyleGAN-generated images. They also introduce a new regularization called path length to enhance the smoothness in the latent space. The intuition behind this regularizer is to ensure that a

fixed-size step in the latent space would result in a non-zero fixed magnitude change in the image.

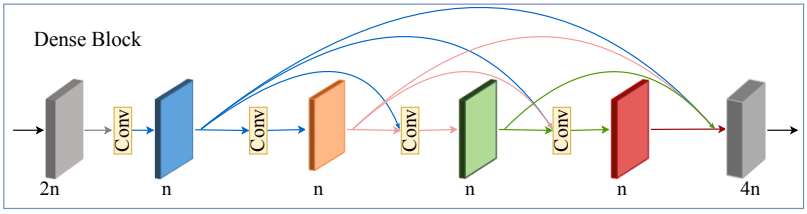
The characteristic artifact observed in the StyleGAN-generated images known as *water droplet effect* is attributed to the AdaIN in the generator. AdaIN is hence replaced with the weight demodulation method in the StyleGAN2 architecture. The weight demodulation method takes the scale and shift parameters in the AdaIN operation out of the sequential computation and introduces the scaling into the weights of the convolutional layers. Another artifact widely observed in the StyleGAN images is the strong location preference for details due to the progressive growing generator and discriminator architecture. As a result, features like teeth or eyes do not move smoothly to the movement in images; instead, they remain stuck in their original position. To overcome these artifacts, StyleGAN2 utilizes an architecture that retains the benefits of progressive growth without its drawback. Inspired by the recent literature in developing better network architectures [98], StyleGAN2 employs an architecture to utilize multiple scales of images via a resnet-style skip connection between low-resolution feature maps to the generated images. The figure 3.1 shows the StyleGAN2 skip generator and residual discriminator. In the generator, the RGB outputs at a different resolution from each stage are up-sampled and then added together to generate the final image. In the discriminator, the down-sampled image and residual connections are provided to each block. An adaptive discriminator augmentation (ADA) technique [100] was later proposed to stabilize the training of styleGAN2 in a limited dataset setting to generate high-quality images. For the CityCount dataset, we consider StyleGAN2-ADA as the base model in our empirical evaluation.

3.3 Multiple Class Count Conditioned Image Generation

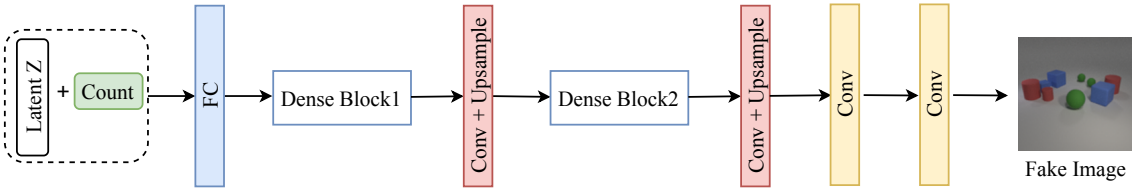
This section introduces the proposed model for multiple-class count-based image generation. We introduce two different models, (1) MC^2-SimpleGAN, a simple network-based architecture of our proposed method for fair and easy comparison study with other conditional GAN variants, and (2) MC^2-StyleGAN2 for state-of-the-art image generation based on the multiple class count.

3.3.1 MC^{2}SimpleGAN

The generator of the MC^2-SimpleGAN is shown in Figure 3.2b. The architecture comprises a count-conditioned generator and a discriminator with an additional count prediction network. Our basic generator network is inspired by the DenseNet [85] architecture. DenseNet



(a) Dense block: Three layer dense block with growth rate n



(b) MC^2-SimpleGAN Generator: A noise sample and a multi-class count vector are passed to a fully connected layer (FC). Two dense blocks (details see above) coupled with conv and two conv. layers follow upsampling layers.

Fig. 3.2 MC^2-SimpleGAN Generator.

introduces dense blocks consisting of several convolutional layers where the output from each layer is connected in a feed-forward fashion to its succeeding layers (see Figure 3.2a for a visualization). The additional skip connections in the dense blocks strengthen the count conditioning in the generator since the input feature maps are connected to the output layers of the dense block [30]. We use two dense blocks of three layers with a growth rate of 64. The generator (Figure 3.2b) gets a combination of randomly sampled noise and multiple class count vectors as input. The concatenated vectors are passed through a fully connected layer (FC) with ReLU activation, followed by dense blocks. The two dense blocks are coupled with a 1×1 convolution to decrease the number of output feature maps and to improve computational efficiency [85] and an upsampling layer to increase the spatial resolution. The output feature maps from the dense block layers are forwarded to two 3×3 convolutional layers (Conv) to generate images. For the discriminator and counting network, we use four convolutional layers with shared weights followed by a fully connected layer to discriminate between real and fake images (discriminator) or to regress the multiple class count vector (count network).

3.3.2 MC^{2}StyleGAN2

We borrow the architectural specifications of the generator and discriminator from StyleGAN2 and extend the model for our application. The input to the generator is a multiple-class

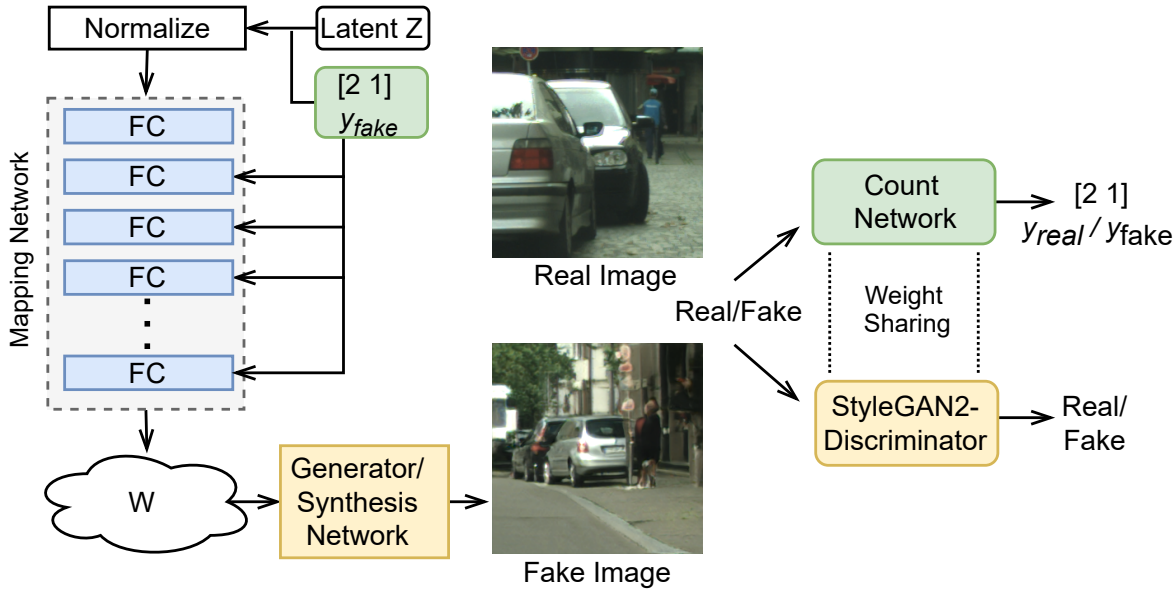


Fig. 3.3 MC^2-StyleGAN2 architecture: The input to the generator is a multiple-class count vector where each vector index corresponds to each object class, and the value at each index represents the multiplicity of the corresponding object class. In the CityCount example, the count vector [2,1] corresponds to 2 cars and 1 person, respectively.

count vector, where each vector index corresponds to a different object class, and the value at each index represents the number of objects from the corresponding object class. The generative part of our model includes a mapping network to map the latent vector and the count constraint to an intermediate latent vector w and a generator/synthesis network to generate images, as shown in Figure 3.3. To the first layer of the mapping network, we provide a combination of randomly sampled noise and our multiple-class count vector, specifying which objects and how many of them are required in the output image. The count vector is also concatenated to every layer in the mapping network, as shown in Figure 3.3. In the generator network, we introduce dense skip connections where the output from each block is connected to its succeeding blocks. As shown in Figure 3.3, the real/generated images are passed through two pathways, (1) an adversarial pathway to classify the input images as real/fake and (2) a count regression pathway to predict the object class and their multiplicity in the input image. The weight sharing between the two sub-networks regularizes the discriminator and reduces memory consumption during training.

3.3.3 Adversarial Training with Count Loss

The generator G, uses both the latent noise distribution $z \sim \mathcal{N}(0,1)$ and a multiple-class count vector $\mathbf{c} = [c_1, c_2, \ldots, c_n]$ that represents n different object classes and their respective multiplicity $c_i, i =, \ldots, n$, to generate fake images $x_{\text{fake}} = G(z, \mathbf{c})$. The discriminator D aims to distinguish between these fake and real images x_{real}. We denote the data distribution as $x \sim p_{data}(x)$. The additional count sub-network C is trained to predict the per-class object count, y_{fake} for fake images, and y_{real} for real images. The adversarial objective of the network is expressed as

$$\mathcal{L}_{GAN}(G,D) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] \; + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z|c)))]. \tag{3.1}$$

The multiple-class count loss $\mathcal{L}_{MC^2}$ is defined as the Euclidean distance between the predicted count $y_{\text{real}} = C(x_{\text{real}})$ and true count $\mathbf{c}$ of the real images, and the distance between the predicted count $y_{\text{fake}} = C(x_{\text{fake}})$ and the value of the count condition for the generated images.

$$\mathcal{L}_{MC^2}(C) = ||C(x) - \mathbf{c}||_2. \tag{3.2}$$

The count loss thus enforces the generator to generate images with the desired number of object instances.

Hence, the total loss of the network is a combination of adversarial loss to match the distribution of real images with fake images and a count loss to enforce the network to generate images based on the specified input count. The overall objective function of our method is,

$$\mathcal{L}_{MC^2-GAN}(G,D) = \mathcal{L}_{GAN}(G,D) + \lambda \mathcal{L}_{MC^2}(C), \tag{3.3}$$

where λ steers the importance of the count objective.

3.4 Experiments and Results

3.4.1 Dataset Used

In this section, we introduce the dataset used for empirical evaluation. We consider the synthetic dataset CLEVR [96], the real dataset MNIST [36] and SVHN [137], and the proposed CityCount dataset. Since multiple class count conditioning of objects is not available for CLEVR, MNIST, and SVHN, we generated a count-conditioned version of these datasets based on the count of objects of interest in both.

Multi-MNIST. MNIST [36] dataset consists of images of handwritten digits. We consider two variants of the MNIST dataset for our experiments, where the images are conditioned based on the number of instances of each digit in the image.

- MNIST-2 with ten digits ranging from 0 to 9 and at most two instances of each digit per image.
- MNIST-3 with ten digits ranging from 0 to 9 and, at most, three instances of each digit per image.

The images were generated by uniformly sampling digits from the MNIST dataset and placing them in non-overlapping positions on black backgrounds. We used 1000 images for each digit combination during training. The count information is provided to the model as a vector with ten entries comprising the desired number of instances of each digit in the image.

Multi-CLEVR. The well-known CLEVR dataset comprises images of different 3D shapes, cylinders, cubes, and spheres of varying colors. For our experiments, we generate a total of 2000 images for each count combination based on the implementation of the CLEVR dataset [96]. We consider two variants of CLEVR images,

- CLEVR-2 with two shapes, cylinder and sphere, and at most six instances of each shape per image. The count label is a vector of 2 entries corresponding to the number of cylinders and spheres in the input image.
- CLEVR-3 with three shapes, cylinder, sphere, and cube, and at most three instances of each shape per image. The count label is a vector of 3 entries corresponding to the number of cylinders, spheres, and cubes in the input image.

For our first line of experiments, we consider a simple setting (CLEVR-Simple), where we restrict shapes of the same class to be of the same color (red cylinders, green spheres, and blue cubes). We extend the experimental setting for further evaluation and consider CLEVR shapes with varying colors.

SVHN-2. We consider real-world images from noisy training data on the street view house numbers (SVHN) dataset [137]. The dataset includes house numbers cropped from street-view images. For our experiments, we considered the original images resized to 64×64 pixels and a total of 1500 samples for each count combination. We restrict ourselves to SVHN images with at most two instances of each digit class (SVHN-2), because images with three or more digits are too scarce for training. The count label is a vector of 10 entries prescribing the multiplicity of each digit in the image.

CityCount To evaluate our method on complex real-world scenarios, we introduce a count-based dataset derived from Cityscapes images, CityCount. Cityscapes [32] dataset was introduced to enhance the semantic understanding of urban scenes. The dataset includes 5000 high quality pixel-level annotated images and 20000 coarsely annotated images of size 1024×2048. The dataset includes 30 different classes and features such as dense semantic segmentation and instance segmentation for vehicle and people classes.

The images in CityCount are collected by cropping 256×256 size patches with a defined number of *cars* and *persons* from Cityscapes. The dataset contains images with at most five instances from each of these classes and roughly 1000 images per object class count combination. To equip our dataset with additional count information, we determine the number of objects per class in each image from the 2D bounding box information of cars and persons from the Cityscapes-3D [56] and the CityPerson dataset [206]. To allow for more diverse appearances of persons in the training set, classes including *pedestrian*, *sitting person*, and *rider* in the Cityscapes images are considered as positive samples when counting the number of persons in the images. This further increases the complexity of the CityCount dataset in terms of spatial arrangement since the network has to infer a reasonable placement of persons, like pedestrians on the sidewalk and riders on the road. Since such additional spatial constraints are not explicitly specified, our dataset is more interesting and challenging for evaluating the proposed approach. Most importantly, the bounding boxes used to generate the training data were not provided to the model during training. The count label is a vector of 2 entries corresponding to the number of cars and persons in the image.

3.4.2 Implementation Details

MC^2-SimpleGAN. The models are trained with images of size 64×64 for Multi-MNIST and SVHN and 128×128 for CLEVR images. All images were scaled at the input with pixel values ranging from -1 to 1. Adam optimizer [105] is used with momentum weights, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ respectively. For the generator and discriminator learning rate is fixed at $1e-4$. The network is trained for 200 epochs with batch size 128 and count loss co-efficient λ as 0.7.

MC^2-StyleGAN2. We extended the official StyleGAN2 TensorFlow implementation [102] corresponding to configuration-e for our count-based image generation for CLEVR and SVHN images. Since the number of training images for CityCount is limited, adaptive discriminator augmentation [100] was applied while training the networks for CityCount images. The mapping network is concatenated with the multiple-class count vector at each layer. We also introduce dense-like connections in place of residual connections in the

(a) Generated MNIST-2 images.

(b) Generated MNIST-3 images.

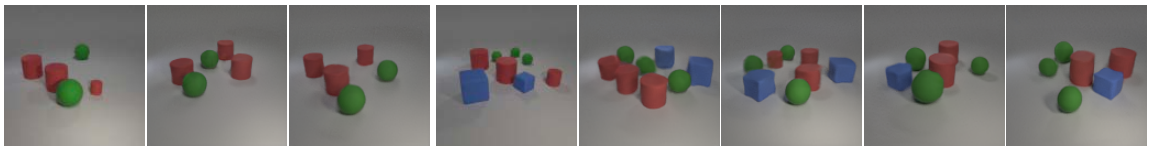
(c) Generated MNIST images for unseen count combination.

Fig. 3.4 Generated Multi-MNIST images for different count combinations - MC^2-SimpleGAN.

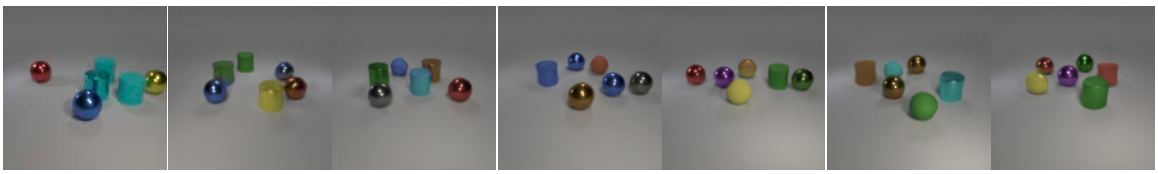
synthesis/generator network for improved results. We fixed the count loss co-efficient λ as 0.8 for all datasets and used the default values from the original StyleGAN2 model for all other hyperparameters. We calculate the FID values on five samples of 50k generated images and report the average value.

3.4.3 Qualitative Analysis

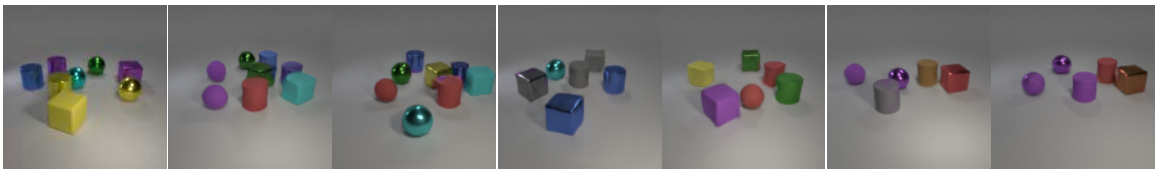
In this section, we visually analyze the quality of the generated images based on the multiple class count vector across the dataset. We start with the simpler version of the proposed model, MC^2-SimpleGAN. We train the network with Multi-MNIST images and evaluate the visual quality of the generated images. Figure 3.4a and Figure 3.4b show the generated MNIST-2 and MNIST-3 images by MC^2-SimpleGAN model for different count combinations. The results show that the proposed model can produce images based on the given digit count without supervision. To check for the interpolation and extrapolation ability of the model, we trained the network with images containing only certain combinations of input count. During testing, we input an unseen count combination of digits (see Figure 3.4c). For the MNIST-2 dataset, the model sees the count value of only 1 for digit 6 during training. Similarly, for the MNIST-3 dataset, the count 3 for digit 9 is unseen during training. The model can transfer the concept of count from one digit to another and predict the count for these unseen



(a) CLEVR-2 and CLEVR-3 (simple) - Count vector corresponds to the number of cylinders, spheres, and cubes.



(b) CLEVR-2 - Count vector corresponds to the number of cylinders and spheres.



(c) CLEVR-3 - Count vector corresponds to the number of cylinders, spheres, and cubes.

(d) SVHN-2 - Count vector corresponds to per digit count.

Fig. 3.5 Generated MC^2-StyleGAN2 images for different count combinations across datasets.

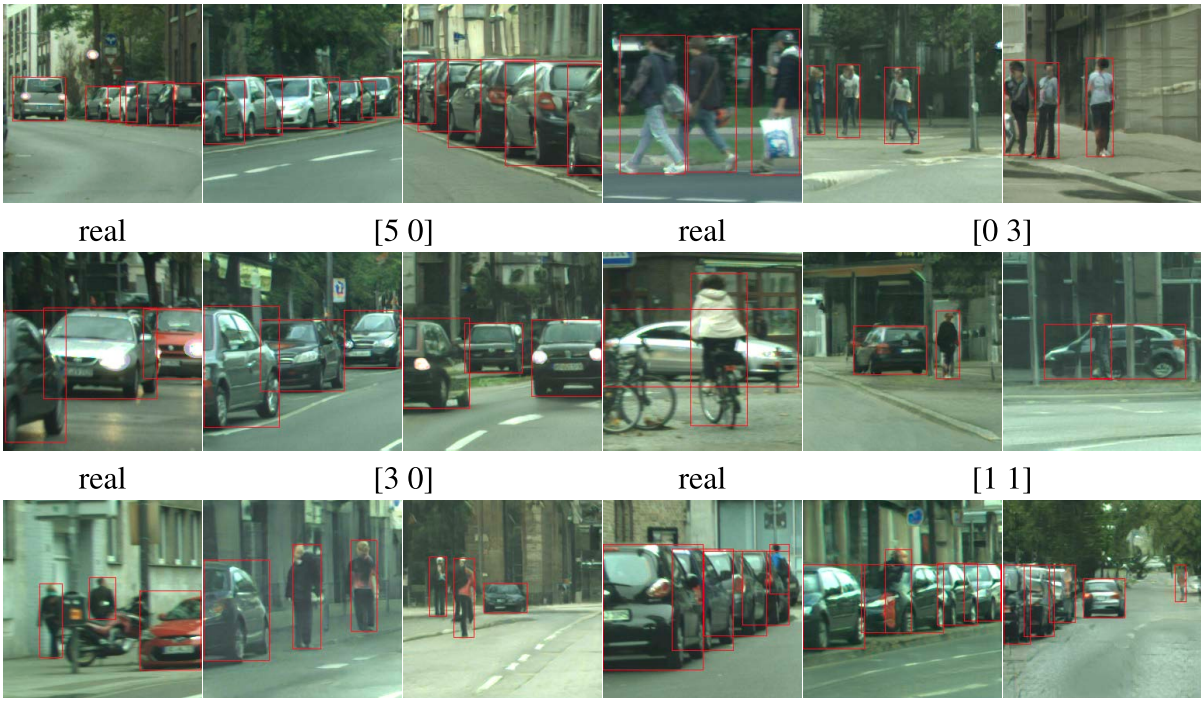


Fig. 3.6 Real and MC^2-StyleGAN2 generated CityCount images - Count vector corresponds to the number of cars and persons. Boxes are drawn around objects of interest for ease of visualization.

combinations. For the count value, 0 for all digits (also unseen during training), the model can generate images without any digits.

For evaluating the MC^2-StyleGAN2 model, we consider Multi-CLEVR, SVHN-2, and CityCount dataset. For Multi-CLEVR images, we start by visually analyzing the generated images for CLEVR-2(simple) and CLEVR-3(simple) as shown in Figure 3.5a. Further, we extend the analysis to complex CLEVR images with varying colors for CLEVR shapes as shown in Figure 3.5b and 3.5c. It can be seen that the model captures the correlation of the count information even in a more complex setting, where the shape colors do not provide additional information. We also observe that the model learned to place objects spatially in reasonable locations, although no object bounding box annotations are provided.

The real and generated SVHN-2 images are shown in Figure 3.5d. The input count vector is of length 10, where each index corresponds to the number of digits from 0 to 9. Although the dataset is noisy and more complex when compared to the CLEVR images, the generated images exhibit higher quality and diversity.

Samples of real and generated CityCount images with their respective count vector are shown in Figure 3.6. Each count vector of size two represents the number of cars and persons. For ease of visualization, boxes are drawn around objects of interest. The model generates images with diverse backgrounds and well-defined person and car classes placed spatially at reasonable locations. As shown in the generated sample of 1 car and 2 people combination in Figure 3.6, the person placed on the road can be seen along with a bike while the second person is standing on the sidewalk. The model learns to distinguish between the pedestrian and the rider class without explicitly defining them in the training set.

3.4.4 Quantitative Analysis

We consider two different metrics to evaluate the performance of the model quantitatively.

- Average count accuracy (Acc) - Evaluates the ability of the model to predict the multiple-class count.
- Fréchet Inception Distance (FID) - Evaluates the quality of the images generated based on the learned count. For more details, please refer to Section 2.2.5.

For count prediction analysis, we consider the performance of the count sub-network in the model. This is defined as the accuracy of the predicted number of objects calculated by rounding the predictions. The quantitative results of our method (MC^2-StyleGAN2) compared to the state-of-the-art conditional GANs such as SNGAN [134], ContraGAN [97] and Conditonal StyleGAN2 [102] are given in Table 3.1. We consistently observed superior

Method	CLEVR-2		CLEVR-3		SVHN-2		CityCount *	
	Acc(↑)	FID(↓)	Acc(↑)	FID(↓)	Acc(↑)	FID(↓)	Acc(↑)	FID(↓)
SNGAN	0.65	40.14	0.61	43.68	0.72	47.34	0.55	55.85
ContraGAN	0.71	25.68	0.68	27.44	0.78	21.12	0.59	49.62
CStyleGAN2	0.70	29.30	0.65	31.95	0.80	19.42	0.61	13.89
Ours	**0.96**	**7.52**	**0.92**	**8.94**	**0.93**	**10.90**	**0.78**	**8.33**

Table 3.1 Quantitative analysis across datasets. *For CityCount we used StyleGAN2 with adaptive discriminator augmentation. [100]

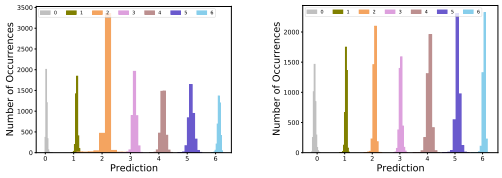
performance in terms of both metrics for our proposed model compared to the baselines across all datasets.

Detailed count prediction analysis. In this section, we analyze the count prediction distribution of individual object classes in CLEVR, SVHN, and CityCount images. The multiple count prediction distribution of the cylinder and sphere class of CLEVR-2 and that of the cylinder, sphere, and cube class of CLEVR-3 are shown in Figure 3.7a and 3.7b respectively. For CLEVR-3, the observed count prediction accuracy is comparatively lower than for CLEVR-2, potentially for two reasons, (1) the image distribution is highly complex due to the high number of objects in the image (maximum of nine objects per image) and (2) objects in the images are often overlapping significantly.

Similarly, the multiple count distribution for the ten-digit classes in SVHN-2 is visualized in Figure 3.7c. We observed an average count prediction accuracy of 93% for SVHN-2 images, with an individual accuracy of 91% for count one and 90% for count two, respectively. We frequently noticed incorrect labels in the original SVHN dataset, which might affect the count label and prediction accuracy.

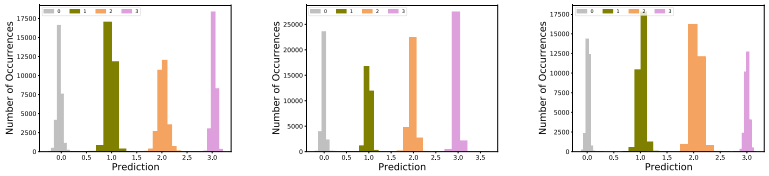
For CityCount images, the predictive performance of the count sub-network for the car and person classes is shown in Figure 3.7d. Here, we compare the predicted count values on the generated samples with the true count provided to the generator network during test time. Since in many samples of the training set, persons are only partially visible and often out of focus or low resolution, we observed a comparatively poor count performance for the person class. For higher counts, 4 or 5, the relatively low performance is presumably due to the lower number of training samples and severe occlusions for the corresponding count.

Interpolation and Extrapolation. We further examine the ability of the model to interpolate between count combinations and to extrapolate to unseen count combinations from one



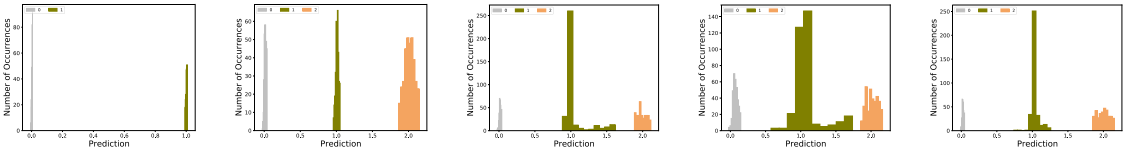
Cylinder Sphere

(a) CLEVR-2

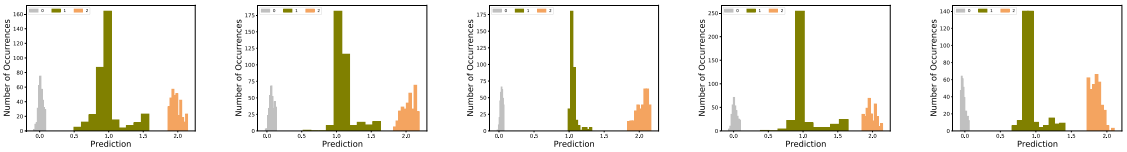


Cylinder Sphere Cube

(b) CLEVR-3

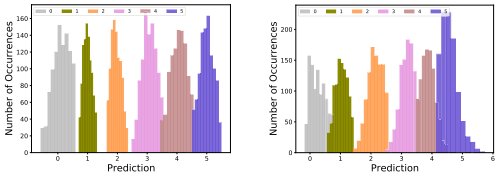


Digit-0 Digit-1 Digit-2 Digit-3 Digit-4



Digit-5 Digit-6 Digit-7 Digit-8 Digit-9

(c) SVHN



Car Person

(d) CityCount

Fig. 3.7 Count performance on Multi-CLEVR, SVHN-2 and CityCount images. The figure shows the predicted count values for each count class.

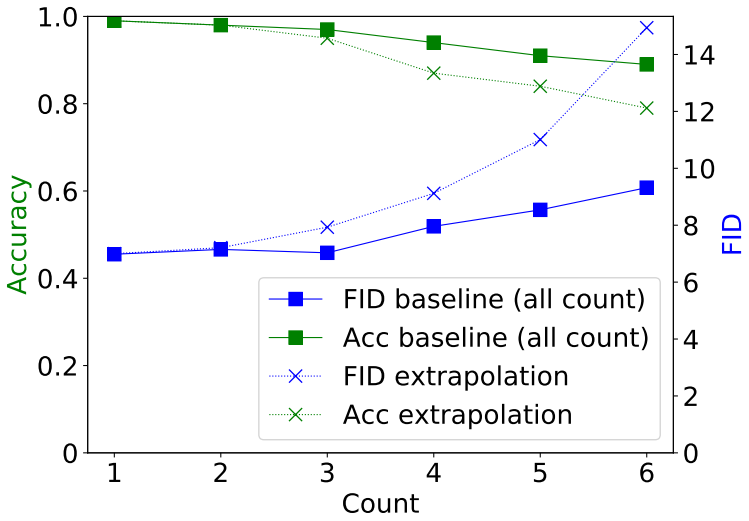


Fig. 3.8 CLEVR-2 extrapolation on spheres based on FID and average count accuracy (Acc). The dotted line indicates the extrapolation performance.

object class to another. To check for interpolation and extrapolation, we train the network with images containing only certain combinations of input count, and during testing, we input an unseen count combination.

For interpolation experiments, we train our model on a subset of CLEVR-2 images that do not contain images with four spheres and a subset of CLEVR-3 without images of two cylinders, while at test time, we evaluate the regression network on exactly such images. The observed count accuracy values for the unseen count during testing are 0.94 and 0.91 for CLEVR-2 and CLEVR-3, respectively. This shows the potential of the model to transfer the learned count four from cylinders to spheres on CLEVR-2 and the learned count two from spheres and cubes to the cylinder class for CLEVR-3 images.

For extrapolation experiments, we train the network with CLEVR-2 images (up to 3 spheres) and plot the success rate in terms of count accuracy and FID to generate 4, 5, and 6 spheres at test time in Figure 3.8. Here the baseline model is trained with images of spheres and cylinders till count 6. The observed extrapolation performance is comparable to the baseline method. This further confirms that the network is not merely memorizing the count number.

Method	Dataset					
	CLEVR-2		CLEVR-3		CityCount	
	Acc(↑)	FID(↓)	Acc(↑)	FID(↓)	Acc(↑)	FID(↓)
w/o Count loss	0.78	18.67	0.80	30.34	0.51	20.24
w/o Discriminator weight sharing	0.91	33.42	0.84	31.03	0.69	15.78
w/o Label mapping	0.90	11.01	0.85	11.32	0.59	8.84
Residual generator	0.94	8.28	**0.93**	11.94	0.65	11.72
Output skip generator	0.94	8.62	0.92	8.98	0.72	10.71
MC^2-StyleGAN2(ours)	**0.95**	**7.98**	0.92	**8.94**	**0.78**	**8.33**

Table 3.2 Ablation study across datasets based on the Average count accuracy (Acc) and Fréchet Inception Distance (FID). The table shows the validity of the proposed architecture choices in our method.

3.4.5 Ablation Study

We perform an ablation study on the synthetic dataset CLEVR and the real dataset CityCount to verify the importance of the additional count loss, generator design, and weight sharing in the discriminator and conditioning methods.

Count loss. We train our model without the count regression network and condition the generator and discriminator with the count label. The rest of our architecture is unchanged. The observed values (w/o count loss in Table 3.2) show that removing the count loss substantially degrades the performance both in terms of count prediction and image quality.

Generator architecture. We consider two different generator configurations introduced in StyleGAN2. One that uses output skip connections and a second one that uses residual connections. As shown in Table 3.2 (residual and output skip generator), our proposed dense-like connections achieve overall good performance in terms of both count prediction and image quality.

Weight sharing in the Discriminator. We compute the evaluation metrics for our model without weight sharing between the count sub-network and the discriminator. The observed values in Table 3 (w/o discriminator weight sharing) show that the model failed to generate the object count correctly. This confirms the positive impact of weight sharing to regularize the count information and inform the discriminator.

Count conditioning in Generator. Lastly, we consider the setting where the count vector is not concatenated to every layer in the mapping network in the generator. Table 3 (w/o label

Method	Dataset					
	CLEVR-2		CLEVR-3		SVHN-2	
	Acc(↑)	FID(↓)	Acc(↑)	FID(↓)	Acc(↑)	FID(↓)
CGAN	0.31	119.23	0.39	186.13	0.39	170.80
InfoGAN	0.37	101.45	0.40	135.36	0.43	151.98
ACGAN	0.38	99.88	0.40	132.23	0.41	150.56
TACGAN	0.40	92.04	0.42	120.11	0.45	138.29
CGAN(ourG)	0.38	88.79	0.45	152.56	0.55	90.34
InfoGAN (ourG)	0.40	75.23	0.44	112.34	0.55	82.13
ACGAN(ourG)	0.41	55.24	0.42	91.02	0.58	70.28
TACGAN(ourG)	0.44	49.01	0.47	87.64	0.61	65.77
MC^2-SimpleGAN(ours)	0.90	47.95	0.89	85.48	0.92	57.52
MC^2-StyleGAN2(ours)	**0.95**	**7.98**	**0.92**	**8.94**	**0.93**	**10.90**

Table 3.3 Comparison with other methods across datasets based on the Average count accuracy (Acc) and Fréchet Inception Distance (FID). Underlined values denotes the proposed method performance on simple (MC^2-SimpleGAN) and bold values with complex architecture (MC^2-StyleGAN2).

mapping) shows that the predictive performance is degraded in this setting. This confirms the benefit of using a count vector-based mapping network to propagate the multiple-class count effectively during training.

3.4.6 Comparison with other Methods

We compare the quantitative performance of other conditional GAN variants, CGAN [133], InfoGAN [27], ACGAN [139] and TACGAN [66], for multiple-class counting on CLEVR and SVHN images. To have a fair comparison of our method with these conditional GAN variants, we use a less evolved network architecture in our proposed model introduced in section MC^{2}SimpleGAN.

The initial results (rows 1 to 3 in Table 3.3) indicate that the considered conditional GAN models did not perform well both in terms of image quality and FID. We even observed mode collapse for CGAN. Hence, we replaced the generator architecture of these models with a Densenet-based generator to improve the performance (rows 4 to 6 in Table 3.3). Although we could greatly improve the initial performance of these models (which shows the positive impact of the proposed Densenet-based generator), MC^2-SimpleGAN clearly

Training data	Acc(↑)		
	CLEVR	CityCount	CityCar
Real only	0.81	0.68	0.77
Real + Aug	0.81	**0.71**	0.78
Real + Syn(ours)	**0.86**	**0.71**	**0.80**

Table 3.4 Average count accuracy across datasets for different training data setting.

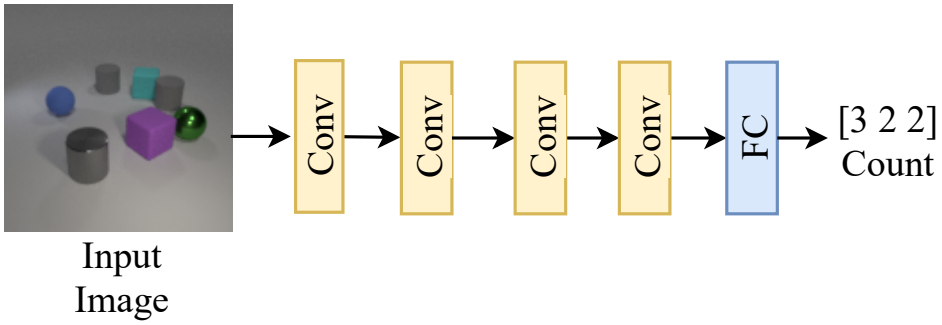


Fig. 3.9 Count prediction network for CLEVR images. The network predicts the number of cylinders, spheres, and cubes in the input RGB images as a count vector

outperforms other methods in the envisioned setting. Further, the quality of the generated images is improved with the proposed MC^2-StyleGAN2.

3.4.7 Training Count Prediction Network using Synthetic Images

We further demonstrate the usability of the images generated by MC^2-StyleGAN2 for training a count prediction network. In particular, we use a multiple-class extension of regression-based architecture similar to the discriminator of MC^2-SimpleGAN. A convolution-based network architecture shown in Figure 3.9 is used for the prediction of multiple-class count prediction of images. The count regression network is similar to the one used in the count sub-network of the MC^2-SimpleGAN. The network includes four convolution layers followed by LeakyRelu activation and dropout layers, with the final block as a fully connected layer that outputs the multiple-class count vector of the corresponding input image.

We design two experiments in this setting using CLEVR and CityCount images. Since the quality of person instances in CityCount images is comparatively low, we also consider a subset of CityCount called CityCar, comprising solely of car class. The average count accuracy of the model is considered the evaluation metric.

Training data	Acc(↑)		
	CLEVR	CityCount	CityCar
Real only	0.81	**0.68**	0.75
Syn(ours) only	0.40	0.30	0.39
25% Real only	0.65	0.41	0.59
25% Real + 75% Syn(ours)	0.67	0.45	0.62
50% Real only	0.76	0.56	0.69
50% Real + 50% Syn(ours)	0.81	0.60	0.75
75% Real only	0.77	0.65	0.74
75% Real + 25% Syn(ours)	**0.83**	**0.68**	**0.76**

Table 3.5 Average count accuracy across datasets when count prediction network trained with real and generated images (Syn) at various proportions.

In the first experiment, we evaluate whether the generated images can improve the count performance when combined with real images during training. For a baseline comparison, the count prediction network is initially trained with real images alone (first row in Table 3.4). The network is then trained with a combination of real and augmented real images (second row in Table 3.4). The observed count accuracy is then compared with the performance of the network when trained with real and generated images (third row in Table 3.4). For a fair comparison, we consider an equal number of augmented and synthetic images. As shown in Table 3.4 for CLEVR and CityCar images, the combination of real and synthetic images (Real+Syn) improved the baseline setting (Real only) and the combination of real and augmented images (Real+Aug). For CityCount, similar count performance is observed for both Real+Aug and Real+Syn.

In the second experiment, we investigate the potential of the generated images to replace the real images during training without compromising the count accuracy performance. We consider the setting where the network is trained with a combination of real and synthetic images at various ratios. Initially, the network is trained with only real images and then with only synthetic images. We gradually replace the real images with synthetic images at various proportions and evaluate the count performance for each setting as shown in Table 3.5. For the baseline comparison of each setting, we consider the count accuracy of the network when trained with the corresponding ratio of the real images only (x% Real only in Table 3.5). As seen in Table 3.5, 50% of real images could be replaced by the generated images without compromising the overall count performance for both CLEVR and CityCar images. The synthetic images could also improve the overall count performance of the network while

replacing 25% of real images for both CLEVR and CityCar images. For CityCount images, 25% of real images could be replaced by the generated images without compromising the overall count performance.

3.5 Conclusion

In this chapter, we investigated the potential of GANs to guide the image generation process based on the number of objects of different classes in the images. While the task of counting is in general very challenging for deep learning approaches, our proposed method can generate images based on the multiple-class count vector in the synthetic and real-world datasets. Our empirical evaluation shows that the model is able to interpolate and extrapolate to unseen counts for specific classes. Even without providing additional information, such as the locations of objects in the image, the network infers a reasonable spatial layout and realization of the objects from the training data distribution solely using the count information.

Chapter 4

Regularized Deterministic Autoencoders

Variational Autoencoders (VAEs) are powerful probabilistic models to learn representations of complex data distributions. One important limitation of VAEs is the strong prior assumption that latent representations learned by the model follow a simple uni-modal Gaussian distribution. Further, the variational training procedure poses considerable practical challenges. Recently proposed regularized autoencoders offer a deterministic autoencoding framework that simplifies the original VAE objective and is significantly easier to train. Since these models only provide weak control over the learned latent distribution, they require an ex-post density estimation step to generate samples comparable to VAEs. In this chapter, we propose a simple and end-to-end trainable deterministic autoencoding framework that efficiently shapes the latent space of the model during training and utilizes the capacity of expressive multi-modal latent distributions. The proposed training procedure provides direct evidence if the latent distribution adequately captures complex aspects of the encoded data. We show in experiments the expressiveness and sample quality of our model in various challenging continuous and discrete domains. This work is published in the Conference on Neural Information Processing Systems (NeurIPS), 2021 [161].

4.1 Introduction

Variational autoencoders (VAEs) constitute one of the popular generative learning frameworks widely used for applications such as image understanding and generation, sentence modeling, and optimizing discrete data and graph-based structures [40, 95, 132, 145, 205]. The VAE framework elegantly combines autoencoders with variational inference [107]. The encoder of the model maps the input data into a lower-dimensional latent space according to a given inference model. The decoder maps the latent space back to the original input space. Both are jointly optimized by maximizing a lower bound on the model evidence, regularizing the

latent space towards a fixed prior distribution, usually a uni-modal Gaussian. By sampling from the latent space prior, we can efficiently utilize the decoder network to generate new samples from the training distribution. Due to the variational formulation, optimizing the VAE training objective poses significant practical challenges. Further, the over-simplistic prior assumption often leads to an unsatisfying trade-off between the quality of reconstructed samples and the prior regularization [14]. Recent work has shown that choosing more flexible priors helps to improve the generative performance of VAEs [179].

Since the initial introduction of VAEs, various novel training objectives have been proposed. One line of work focuses on different regularization techniques derived from alternative probabilistic metrics to shape the latent space of the model during training, e.g., using the Wasserstein distance [178]. In contrast to the KL divergence, the Wasserstein distance measure induces a metric on probability distributions. Practically, this facilitates smoother convergence even for initially non-overlapping distributions. Further, it overcomes the over-regularization effect in VAEs. To be precise, it prevents the undesired behavior of multiple data points from being mapped to the same latent representation by the encoder. Since closed-form solutions for metrics like the Wasserstein distance can only be derived for very few prior distributions, these approaches rely on numerical approximations during training.

Recent work by Ghosh et al. [60] reinterprets deterministic autoencoders as variational models, even when trained with a deterministic loss. During training, this approach maximizes the negative log-marginal likelihood of the latent samples under a Gaussian normal distribution and minimizes the reconstruction loss. Experimental results show that this regularization alone does not suffice to generate high-quality samples using the Gaussian prior. To overcome this, Gosh et al. propose to use a multi-modal Gaussian mixture model (GMM) to fit arbitrary, learned latent spaces. While this approach leads to good sampling efficiency and generalization if the post-hoc fit is reasonable, sampling quality can suffer significantly if the learned latent space can not be modeled well by a GMM.

In this chapter, we propose a deterministic training scheme for autoencoders that applies to expressive priors and overcomes the necessity of a post-hoc density estimation step for deterministic training. To be precise, we derive a deterministic regularization loss from the distance metric used in the non-parametric Kolmogorov-Smirnov (KS) test for equality of probability distributions. The resulting training objective can be derived in closed form for a class of expressive multi-modal prior distributions and provides a strong signal to efficiently shape the model's latent space during training. We chose our experiments to evaluate the proposed approach regarding sampling quality and expressiveness. In the first line of experiments, we compare the quality of newly generated and reconstructed samples

from our model with those from various VAE variants. In the second line, we investigate our method's capability to model discrete and complex structured inputs such as arithmetic expressions and molecules. VAEs have recently been proposed in these domains as a tool for dimensionality reduction in optimization. Applying our regularization scheme effectively utilizes multi-modal prior distributions in this context and significantly improves optimization performance.

4.2 Related Work

Since our proposed regularization objective structures the latent space to a Gaussian mixture model, we also compare it to prior work on deep clustering. Next, we discuss VAEs in the context of black-box optimization approaches such as Bayesian Optimization (BO).

Deep Clustering Deep Clustering approaches benefit from well-structured latent spaces. Thus, several methods employ Gaussian mixture VAEs for data encoding [39, 144] or establish a GMM-like latent space structure through k-means models in the latent space. For example, Xie et al. [200] train an autoencoder and apply a KL-divergence loss for better k-means clustering, while Ghasedi et al. [43] combine the autoencoder reconstruction loss with the relative cluster entropy. Similar approaches have been proposed in the literature [43, 72, 77, 92, 176, 202]. Caron et al. [130] iteratively group points using k-means during optimization.

Structural Variational Autoencoders and Optimization High-dimensional optimization problems in structured discrete input domains are ubiquitous. VAEs have been used in this context to learn low-dimensional, continuous representations of high-dimensional, structured data like molecules or arithmetic expressions. Recent work proposes to use such representations to perform efficient optimization by running BO in the latent space of VAEs [114, 122]. In this setting, prior knowledge of the structure of the latent space is crucial to allow for an efficient exploration and generation of valid samples. Yet, as discussed above, VAEs can suffer from simplistic prior assumptions. Thus, sampling from the latent space of such models can result in invalid samples, reducing the sampling efficiency of BO [73]. Kusner et al. [114] overcome this issue if data follows a specific grammar. Lu et al. [122] propose a VAE that directly works on parse trees from context-free grammars to represent discrete data. Yet, those only work with unimodal priors, limiting generalization capabilities. Our approach can be readily used to extend these models to encode structural data better and improve BO performance.

4.3 Regularization in Deterministic Autoencoders

We introduce a novel loss function to regularize the latent representation learned by deterministic autoencoders towards a given prior distribution. The definition of our loss builds on the non-parametric statistical Kolmogorov-Smirnov (KS) test for equality of one-dimensional probability distributions. We propose a multivariate variant of the distance measure used in the KS test that allows for gradient-based optimization and can easily be applied to expressive multi-modal prior distributions. For ease of exposition, we introduce our regularization loss for unimodal Gaussian priors in section 4.3.1 and extend the formulation to expressive multi-modal Gaussian mixture models in Section 4.3.2. Finally, in Section 4.3.3, we provide an explicit way to estimate the weighting parameters of our loss.

4.3.1 Uni-Modal Latent Regularization

The KS test can be used to determine whether a collection of N, one-dimensional samples follow a given reference distribution. It compares the cumulative distribution function (CDF) of the reference distribution with the empirical CDF $\bar{F}^{(N)}$ of the samples. It is often applied to one-dimensional Gaussian distributions, which have important analytical properties. For spherical Gaussians, the one-dimensional KS test quantifies the distance between the empirical distribution function of the data and the cumulative distribution function

$$\Phi(z) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp \frac{-(t-\mu)^2}{2\sigma^2} dt \tag{4.1}$$

of the univariate Gaussian $Z \sim \mathcal{N}(\mu, \sigma)$ as $\sup_{z\in\mathbb{R}} |\bar{F}^{(N)}(z) - \Phi(z)|$. Extending this KS distance to a higher dimension is particularly challenging since it requires matching joint CDFs [52, 69, 143]. Especially in higher dimensions, this becomes infeasible [121]. The continuously ranked probability score [63] shares the same theoretical basis as the KS distance. However, it tests whether two sets of samples are consistent with each other, i.e., they could originate from the same distribution and is thus not suitable to regularize a collection of latent samples towards a given prior distribution. Alternative multi-variate normality tests, like the Mardia test [128] and the BEHP test [10] suffer from slow convergence rates.

To derive a regularization loss from the KS distance, we propose to overcome this issue by separately considering the marginal CDFs and correlations in the prior distribution. Given d-dimensional latent samples $\mathbf{z}_1, \ldots, \mathbf{z}_N$, the empirical marginal CDF in dimension j is given

by

$$\bar{F}_j^{(N)}(z) = \frac{1}{n}\sum_{n=1}^{N} \mathbb{1}_{[\mathbf{z}_n]_j \leq z}. \tag{4.2}$$

We aim to regularize the latent space of our models by comparing the empirical marginal CDFs with the one-dimensional CDFs of the marginal distributions of the prior. To strengthen the training signal of our regularization scheme and make it suitable for gradient-based optimization, we replace the supremum in the original KS distance with a smoother MSE loss that compares the distances between those functions at the latent representations. For a uni-modal Gaussian prior with mean μ and covariance matrix Σ, this results in

$$\mathcal{L}_{\mathrm{KS}}(\mathbf{z}_{1,\dots,N}) = \frac{1}{d}\sum_{j=1}^{d} \mathrm{MSE}\left(\bar{F}_j^{(N)}(\mathbf{z}_j), \Phi(\bar{\mathbf{z}}_j)\right), \qquad \bar{\mathbf{z}}_j = \frac{\mathbf{z}_j - \mu_j}{[\Sigma]_{j,j}}. \tag{4.3}$$

Here, $\bar{F}_j^{(N)}(\mathbf{z}_j)$ denotes the vector with entries $\bar{F}_j^{(N)}([\mathbf{z}_i]_j)$ and $\Phi(\bar{\mathbf{z}}_j)$ is defined accordingly. This loss is minimized if the empirical marginal CDFs of the latent samples match those of the uni-modal Gaussian prior. Using the above loss alone will not account for correlations between different latent dimensions. In the case of a spherical Gaussian prior with an identity covariance matrix, for example, samples with perfectly correlated Gaussian components $[\mathbf{z}_i]_j = [\mathbf{z}_{i'}]_j$, will also minimize this objective, see Figure 4.1. To overcome this problem, we equip our loss with an additional term that matches covariances between different latent distributions explicitly. Following a similar reasoning to the MSE above, we define an additional loss term,

$$\mathcal{L}_{\mathrm{CV}}(\mathbf{z}_{1,\dots,N}) = \frac{1}{d^2}\sum_{l,j=1}^{d} \left([\bar{\Sigma}]_{l,j} - [\Sigma]_{l,j}\right)^2, \tag{4.4}$$

where $\bar{\Sigma}$ is the empirical covariance matrix of the latent representations and Σ stands for the prior covariance. Compared to the negative log marginal regularization proposed in [60], our loss will enforce the latent representations to be spread across the entire support of the Gaussian prior instead of being minimal when all latent collapse to the origin.

4.3.2 Multi-Modal Latent Regularization

One advantage of our approach is the applicability to more expressive, multi-modal prior distributions. While the Gaussian distribution has important analytical properties, it suffers from significant limitations when modeling real data sets. In contrast, a linear combination of

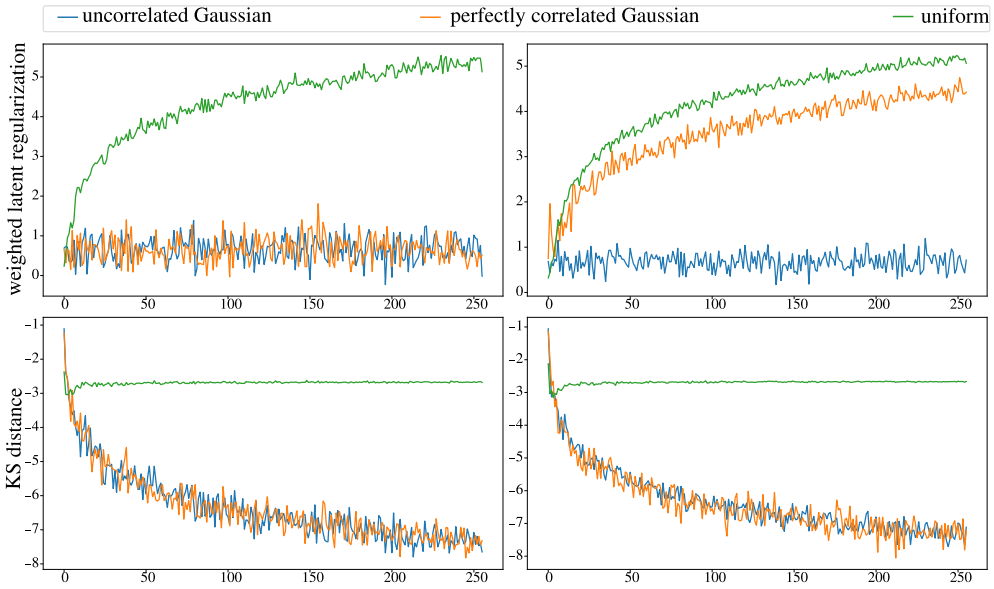


Fig. 4.1 Uni-modal latent regularization in one and two dimensions for varying numbers of samples (x-axis) from different distributions: In two dimensions (right), the simplistic KS distance can not differentiate the target prior (blue) from other probability distributions. By contrast, our proposed regularization scheme successfully matches correlations across different dimensions.

Gaussians can produce very complex densities while allowing for closed-form computations of important quantities, like CDFs and covariances. A d-dimensional K-modal Gaussian mixture model is a weighted superposition of K Gaussian distributions in $\mathbb{R}^d$, often referred to as the modes of the model. For $k \leq K$, let μ_k and Σ_k be the mean and covariance matrix of the k-th mode in the model. Further, let $p_k > 0$ be the weight of the k-th mode. Then, the marginal CDFs of a GMM model can be computed from the CDFs of univariate Gaussians as follows

$$F_{\text{GMM},j}(z) = \sum_{k=1}^{K} p_k \Phi\left(\frac{z - [\mu_k]_j}{[\Sigma]_{j,j}}\right), \tag{4.5}$$

i.e., the marginal CDFs in the GMM are weighted sums of CDFs of one-dimensional Gaussians. The covariance matrix of the GMM can be computed as

$$\Sigma_{\text{GMM}} = \sum_{k=1}^{K} p_k \Sigma_k + \sum_{k=1}^{K} p_k \left(\mu_k - \bar{\mu}\right)\left(\mu_k - \bar{\mu}\right)^T, \qquad \bar{\mu} = \frac{1}{k}\sum_{k=1}^{K} \mu_k. \tag{4.6}$$

Extending our proposed regularization scheme to multimodal GMMs is straightforward. Our first loss term is defined as

$$\mathcal{L}_{\mathrm{KS},K}(\mathbf{z}_{1,\dots,N}) = \frac{1}{d}\sum_{j=1}^{d} \mathrm{MSE}\left(\bar{F}_j^{(N)}(\mathbf{z}_j), F_{\mathrm{GMM},j}(\mathbf{z}_j)\right). \tag{4.7}$$

Similarly, the second loss term is defined to be

$$\mathcal{L}_{\mathrm{CV},K}(\mathbf{z}_{1,\dots,N}) = \frac{1}{d^2}\sum_{l,j=1}^{d} \left([\bar{\Sigma}]_{l,j} - [\Sigma_{\mathrm{GMM}}]_{l,j}\right)^2. \tag{4.8}$$

The total loss of the model is a combination of the reconstruction loss and a regularization loss that enforces the latent representations of the encoded data to match a predefined multimodal prior distribution. The reconstruction loss $\mathcal{L}_{\mathrm{REC}}(\mathbf{x}'_{1,\dots,N})$ equals the mean squared error between inputs $\mathbf{x}_i$ and their reconstructions $\mathbf{x}'_i$. Given positive weights λ_{KS} and λ_{CV}, our final loss is given by

$$\mathcal{L}(\mathbf{x}_{1,\dots,N}) = \lambda_{\mathrm{REC}}\mathcal{L}_{\mathrm{REC}}(\mathbf{x}'_{1,\dots,N}) + \lambda_{\mathrm{KS}}\mathcal{L}_{\mathrm{KS},K}(\mathbf{z}_{1,\dots,N}) + \lambda_{\mathrm{CV}}\mathcal{L}_{\mathrm{CV},K}(\mathbf{z}_{1,\dots,N}). \tag{4.9}$$

Formally, the weights $\lambda_{\mathrm{KS}}, \lambda_{\mathrm{CV}}$ and λ_{REC} are hyperparameters of the model. Nevertheless, we propose an explicit way to set λ_{KS} and λ_{CV} and a simple heuristic to estimate λ_{REC} to avoid an extensive optimization of these weights.

4.3.3 Loss weight estimation

Balancing the two regularization losses appropriately poses a key challenge, as they vary on different scales. For example, if modes of the GMM prior are far spread, the covariance $\mathcal{L}_{\mathrm{CV},K}$ loss will dominate the marginal CDF $\mathcal{L}_{\mathrm{KS},K}$ loss by far. Nevertheless, given a target GMM prior, the dimension of the latent space, and the batch size n used during training, there is a concise way to fix those hyperparameters beforehand. To be precise, for $m = 1, \dots, M$ samples $\mathbf{z}_1^{(m)}, \dots, \mathbf{z}_N^{(m)}$ from the prior GMM, we propose to set

$$\lambda_{\mathrm{KS}}^{-1} = \frac{1}{M}\mathcal{L}_{\mathrm{KS}}\left(\mathbf{z}_{1,\dots,N}^{(m)}\right), \qquad \lambda_{\mathrm{CV}}^{-1} = \frac{1}{M}\mathcal{L}_{\mathrm{CV}}\left(\mathbf{z}_{1,\dots,N}^{(m)}\right). \tag{4.10}$$

Formally, we can not overcome the necessity of tuning the weight of the reconstruction loss, which significantly impacts the model's performance. Nevertheless, a reasonable approximation to it can be obtained by training an autoencoder model and using the inverse

of the best obtained loss for λ_{REC}. Using this scaling, all loss terms in our regularization loss will ultimately converge to one of the targets prior is matched successfully.

4.4 Experiments and Results

With our experiments, we strive to investigate the potential of the proposed model compared to other VAE variants in generating new samples, analyze the effect of the chosen prior in clustering the latent space, and shape the latent space efficiently in highly structured domains such as discrete spaces.

4.4.1 Analysis of the Proposed Latent Regularization

This section analyzes the unimodal and multimodal versions of the proposed regularization loss across different distributions.

Unimodal Regularization loss. For a fixed target prior, we investigate the behavior of our loss on samples from varying distributions. Throughout the unimodal analysis, we choose a standard normal distribution as the target prior. In our experiments, we evaluate our loss on the following settings,

- Samples from a Unimodal Gaussian distribution with standard deviation equal to the prior, but different mean.
- Samples from a Unimodal Gaussian distribution with mean equal to the prior, but varying standard deviation.

The observed values for the proposed weighted regularization loss are plotted in Figure 4.2 for dimensions 1 and 2. It can be observed that the loss function increases with increasing distance between the means and standard deviations of the sampling distribution and the target prior.

Multimodal Regularization loss. Throughout the multimodal analysis, we fix a Gaussian mixture model as the target prior with two equally weighted spherical components centered at one hot encoding vector of the respective dimensions. Similar to the above, we consider two sets of experiments for the evaluation:

- Samples from a GMM with two spherical components centered at different means.

- Samples from a GMM with two components centered at the means of the prior components but different covariance.

In both cases, we vary the means and covariances by adding a multiplicative factor α and β to the means and covariance matrices of the prior, respectively. The observed values for the proposed weighted regularization loss are plotted in Figure 4.3 and Figure 4.4 for dimensions 2 and 3. It can be observed that the value of the loss function increases with the increasing factors α and β.

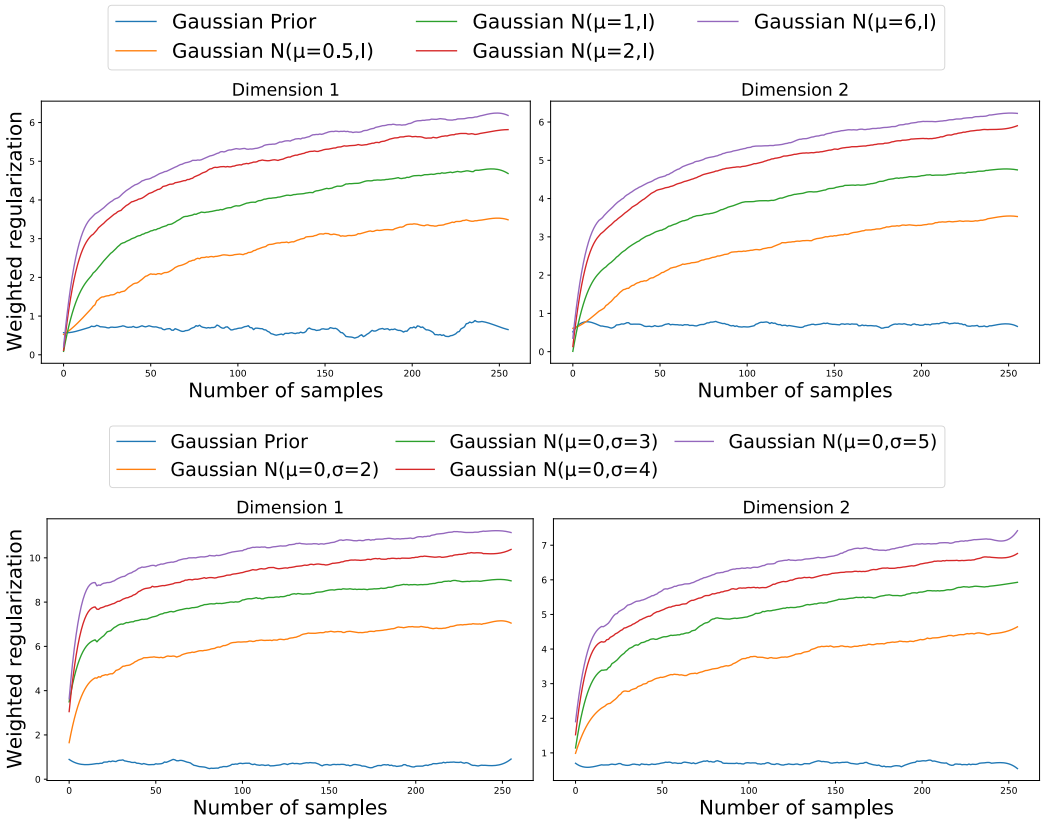


Fig. 4.2 Loss analysis - Unimodal latent regularization in one and two dimensions for varying numbers of samples from different Gaussian distributions. With the increase in mean and standard deviation, the loss function increases with respect to the target prior (blue).

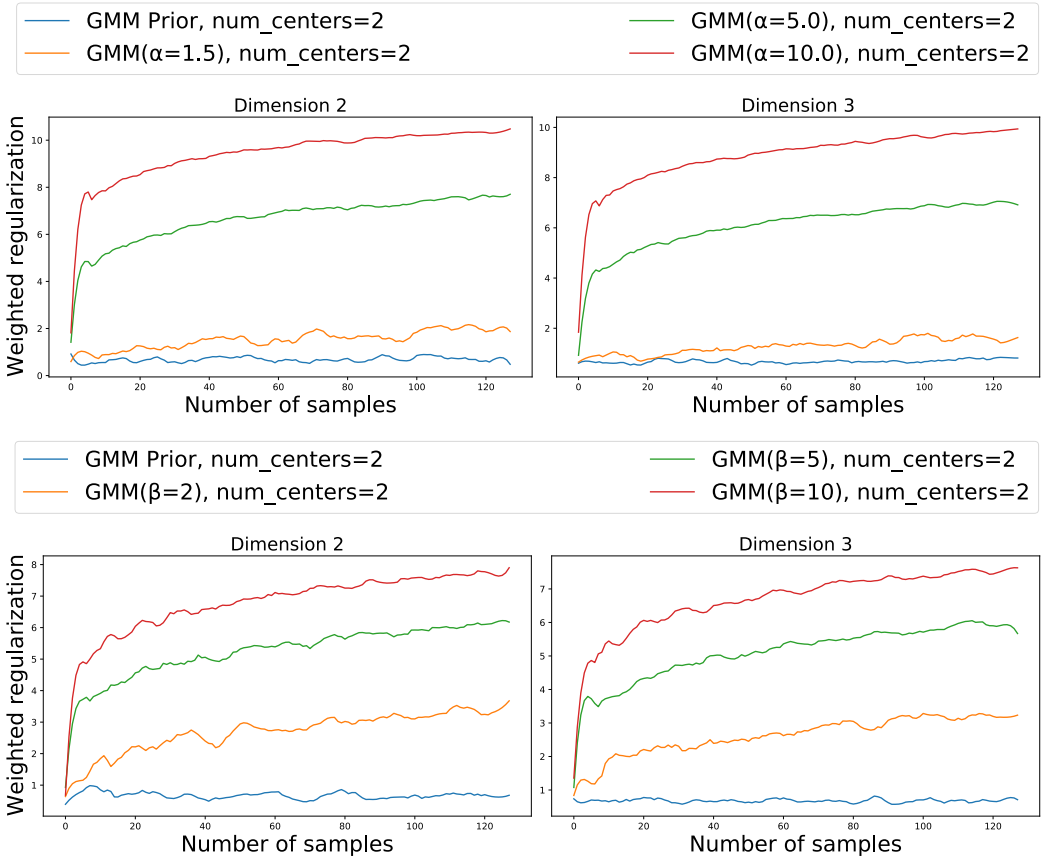


Fig. 4.3 Loss analysis - Multimodal latent regularization in two and three dimensions for varying samples from different Gaussian mixture distributions. With an increase in the mean and covariance of the samples, the loss function increases with respect to the target GMM prior (blue).

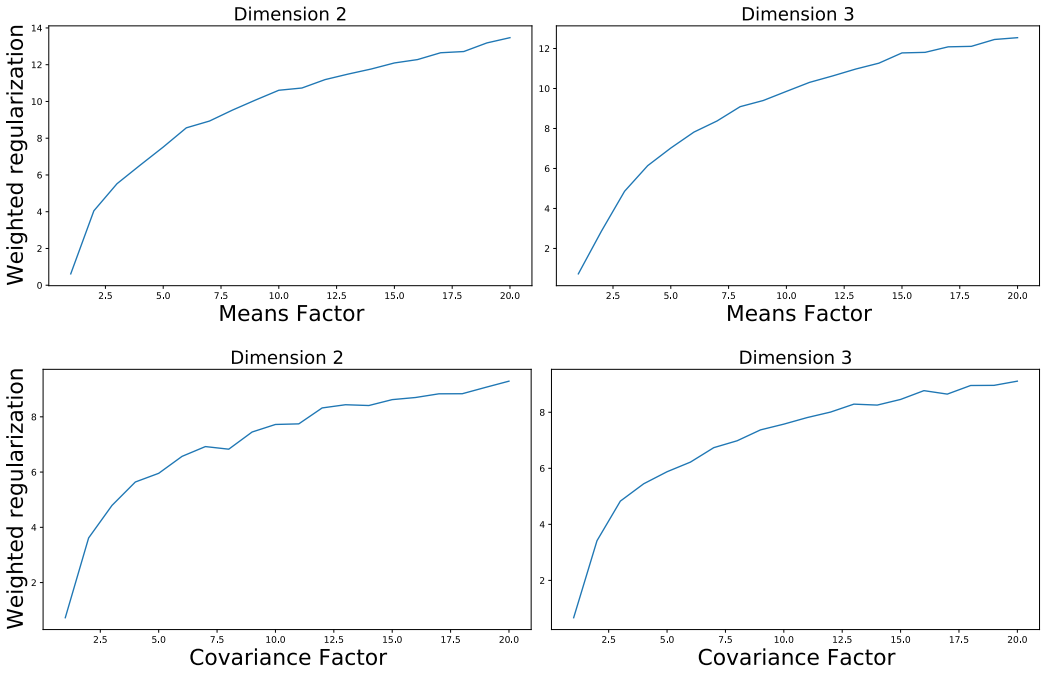
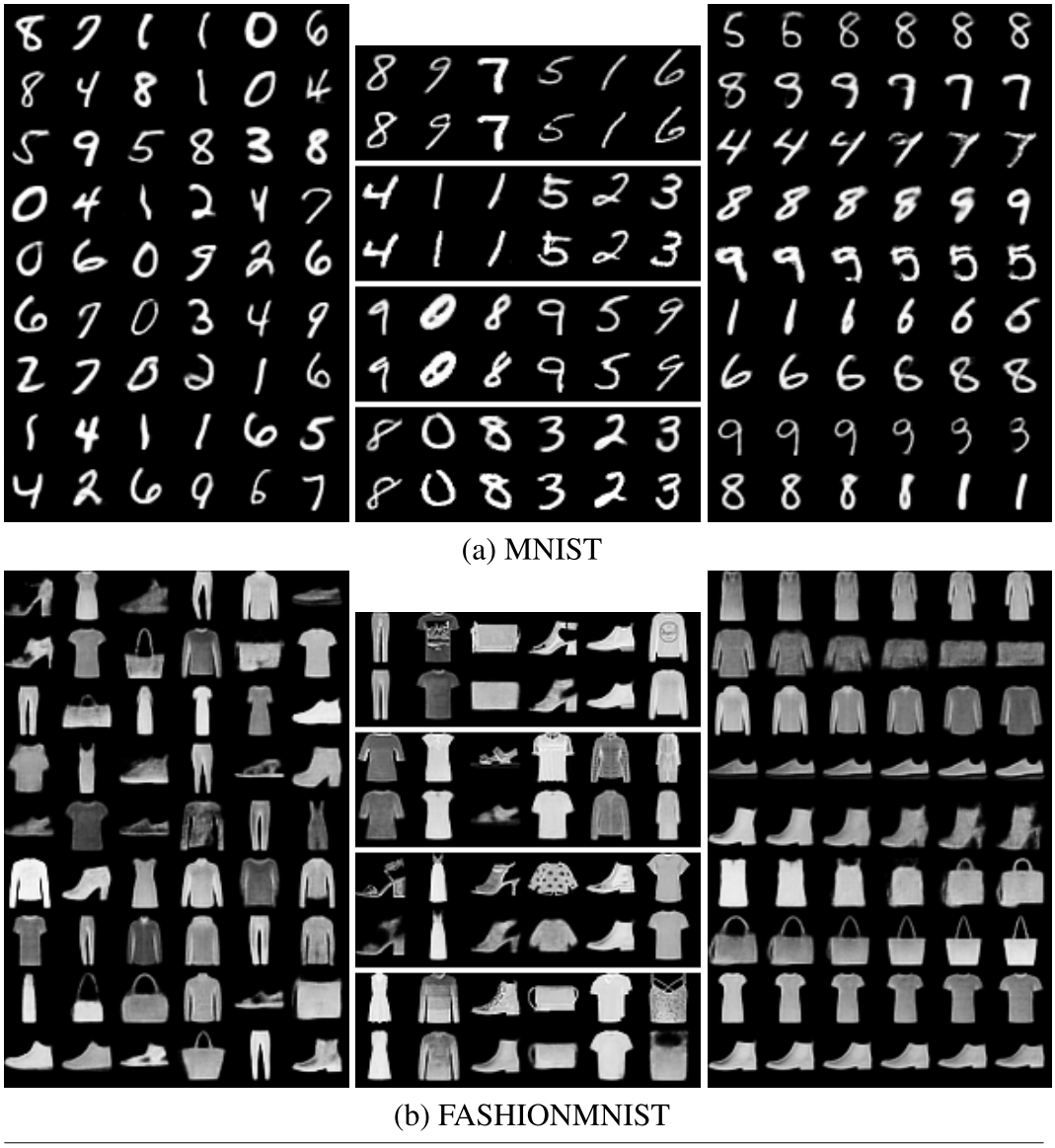


Fig. 4.4 Loss analysis - Multimodal latent regularization in two and three dimensions for the different mean (α) and covariance factor (β).

4.4.2 Image Generation

We consider four dataset, MNIST [36], FASHIONMNIST [199], SVHN [137] and CELEBA [120] to evaluate the proposed method in image generation experiments. The qualitative analysis of the generated samples for MNIST, FASHIONMNIST, SVHN, and CELEBA images are shown in Figure 4.5 and Figure 4.6. along with the reconstructed and interpolated samples in the latent space of the trained model. To assess the quality of the generated images, we evaluate the Fréchet Inception Distance (FID) [78] for each dataset, see Table 4.1. For a baseline comparison, we evaluate the following models: vanilla variational autoencoder (VAE [107]), Gaussian mixture variational autoencoder (GMVAE) [39], Wasserstein autoencoder (WAE) [178] with MMD loss, 2stage VAEs (2s-VAE) [34], constant variance-VAE (CV-VAE) [60] and regularized autoencoders (RAEs) [60]. We consider the following evaluation metrics: 1. Sampling FID (Samp.) - FID score of the generated random samples (evaluated by generating random samples from the prior distribution of the respective models and by fitting a Gaussian distribution to models trained without any prior assumptions), 2. reconstruction FID (Rec.) - measured by computing the FID between the test samples and their corresponding reconstructions by the model and 3. interpolation FID (Inter.) - measured by computing the FID between the interpolated samples in the latent space and test samples. As pointed out by [60], fitting an ex-post density estimator on the learned embedding after training VAEs further improves the generation quality. Hence, we also report the FID values by fitting a GMM in the learned latent space of the trained model (GMM column in Table 4.1, not evaluated for 2s-VAE as they perform ex-post density estimation using another VAE).

As shown in Table 4.1, our method achieves better FIDs (Samp.) on all datasets compared to all baselines sampled by fitting a single Gaussian in the latent space. As argued above, we also improved the generation quality by fitting a mixture of Gaussians in the latent space and achieving better FIDs in MNIST, FASHION MNIST, and CELEBA images. In contrast, for SVHN, WAEs achieved the best score. It is also important to note that the proposed method performs comparably or even better without employing the ex-post density estimation. The proposed method achieves better reconstruction quality than the other VAEs, except for SVHN images, where RAEs perform better. The interpolation FID indicates the overall structure of the learned latent space. The obtained FID values show that the proposed method shapes the latent space better than the other approaches, except for the CELEBA images, where RAEs perform slightly better than ours. We use the same architecture and experimental settings in all the considered baseline evaluations for a fair comparison. Please refer to the Appendix for more details on the experimental settings.



(a) MNIST

(b) FASHIONMNIST

Fig. 4.5 Qualitative analysis on image generation across MNIST and FASHIONMNIST. Column 1 shows the randomly generated samples; column 2 shows the reconstructed samples by the decoder on the test dataset after training (the first row in each section corresponds to the ground truth and the second one its corresponding reconstruction), and column 3 shows randomly interpolated samples in the learned latent space of our model.

(a) SVHN

(b) CELEBA

Fig. 4.6 Qualitative analysis on image generation across SVHN and CELEBA images. Column 1 shows the randomly generated samples; column 2 shows the reconstructed samples by the decoder on the test dataset after training (the first row in each section corresponds to the ground truth and the second one its corresponding reconstruction), and column 3 shows randomly interpolated samples in the learned latent space of our model.

Dataset	MNIST				FASHION MNIST			
	Samp.	GMM	Rec.	Inter.	Samp.	GMM	Rec.	Inter.
VAE	27.27	20.52	21.59	21.05	50.50	36.22	33.33	44.12
GMVAE	21.35	–	20.64	20.21	40.23	–	38.79	38.54
WAE	20.20	12.90	14.07	16.19	39.66	28.01	24.84	35.01
CV-VAE	32.12	28.62	29.61	30.76	57.57	38.28	35.10	47.73
2sVAE	26.99	–	23.77	22.13	46.47	–	31.93	41.06
RAE	17.72	14.15	14.69	15.57	47.26	29.59	24.54	34.77
Ours	**13.11**	**12.82**	**8.99**	**12.82**	**33.70**	**26.62**	**19.56**	**29.17**
Dataset	SVHN				CELEBA			
	Samp.	GMM	Rec.	Inter.	Samp.	GMM	Rec.	Inter.
VAE	61.01	58.23	59.13	50.29	68.01	61.63	52.55	58.39
GMVAE	49.74	–	48.65	47.15	65.35	–	64.22	64.92
WAE	58.08	**34.87**	29.62	27.16	58.91	49.17	41.14	47.08
CV-VAE	51.01	54.19	48.53	47.65	57.61	52.72	45.32	50.87
2sVAE	45.84	–	44.27	40.23	53.12	–	44.78	47.64
RAE	42.35	35.12	**31.04**	27.30	52.33	48.23	41.61	**46.58**
Ours	**37.42**	36.46	31.27	**24.87**	**49.79**	**44.79**	**39.48**	47.13

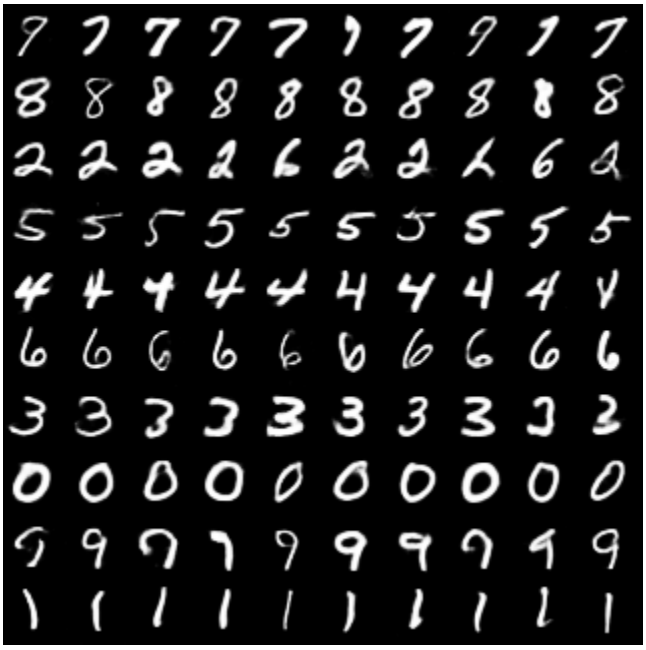
Table 4.1 Quantitative evaluation results across datasets. Samp. refers to the FID of the generated samples from the prior distribution or by fitting a Gaussian to the learned models trained without prior. GMM refers to the FID computed by fitting GMM on the learned model, Rec. refers to the reconstruction FID on test samples, and Inter. refers to the Interpolation FID.

4.4.3 Unsupervised Image Clustering

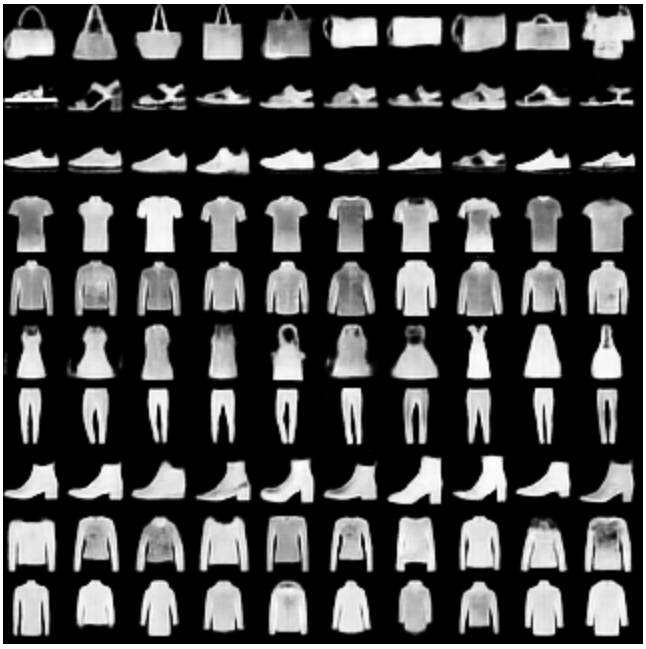
We evaluate the potential of our method to naturally cluster the data points in the learned latent space in two datasets, MNIST and FASHIONMNIST. The Gaussian mixture model prior with k components in our method could be considered as k different classes/clusters to which the encoder maps the data points. We train the model with latent space dimension 10 for both datasets and visualize the random samples generated from each Gaussian component of our prior as shown in Figure 4.7. The figure shows that visually similar images fall into the same cluster. For a quantitative analysis of the clustering performance, we evaluated the unsupervised classification accuracy (similar to [91]) and compared the performance with JointVAE [48] and CascadeVAE [91]. The observed values are reported in Table 4.2. We observed a comparable performance to both baselines. We also observed that the distance between the modes in the GMM prior is a deciding factor in better clustering performance. Figure 4.8 shows the performance comparison of both image generation and clustering performance with increasing distance between different modes in the GMM prior. The result shows that with increasing distance, the clustering performance is improved, whereas the quality of the generated images gets reduced. we also report the unsupervised clustering performance in terms of two other metrics, 1. Normalized Mutual Information (NMI) and 2. mean Average Precision (mAP). NMI measures the mutual information between the cluster assignments and the ground truth labels and is normalized by the average of the entropy of both target and observed labels. The calculated NMI and mAP values for the MNIST are 0.72 and 0.75, and for the FASHION MNIST, 0.60 and 0.61, respectively. Our experimental analysis indicates that natural clustering happens with the multi-modal GMM prior.

Method	Acc(↑)	
	MNIST	FASHIONMNIST
JointVAE	78.33	51.51
CascadeVAE	84.19	**57.72**
Ours	**85.53**	56.24

Table 4.2 Unsupervised classification results on MNIST and FASHIONMNIST images.



MNIST



FASHIOMNIST

Fig. 4.7 Clustering performance on MNIST and FASHIONMNIST images with a 10 component GMM prior. Each row in the figure shows randomly generated images from different Gaussian components of the GMM prior. Similar looking images are mapped into the same clusters.

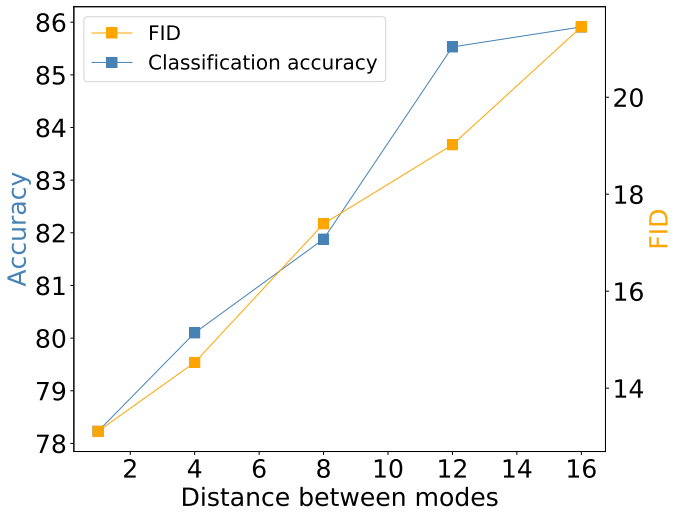


Fig. 4.8 Image clustering(Accuracy) and generation performance(FID) on MNIST images with an increase in the distance between modes in the GMM prior.

4.4.4 Modelling Discrete Data Structures

In this section, we investigate the ability of our model to generate complex discrete data structures such as arithmetic expressions and molecules. This experiment aims to analyze the model performance in effectively shaping the latent space of such structured discrete spaces. The learned latent space of the model is traversed to generate new samples with the desired properties by performing Bayesian Optimization (BO). We perform experiments in two sequence optimization problems similar to [114].

Arithmetic Expression Given a dataset of $50,000$ univariate (functions of x) arithmetic expressions following a formal grammar [114], the task is to find the expression that best fits a target dataset. This is done by minimizing $\log(1+\text{MSE})$, where the MSE is computed between values of the generated expression and the target points. We choose similar target data points for our evaluation as in [114].

Chemical Design Given the ZINC250k dataset of drug molecules [88], the objective is to generate new drug like molecules. The drug likeliness of a molecule is quantified by the water-octanol partition coefficient, maximized in our line of experiments.

Results We extend the architecture and experimental settings of [114] to include our proposed losses during training. For baseline comparison, we consider Grammar VAE (GVAE) [114],

Character VAE (CVAE) [73], Grammar constant variance VAE (GCVVAE) [60] and Grammar based RAE (GRAE) [60] frameworks. The three best scores found by our method for arithmetic expressions and the molecule experiments are reported in Table 4.3. Our model performs comparatively better than the considered baselines and achieves the best first score for both tasks. In addition to the optimization performance, it is also important to consider the validity of the new samples generated by the models. A well-structured latent space should yield valid samples following the defined grammar/rules of the used dataset. Our model achieves better validation and average scores as shown in Table 4.4 except for GCVVAE, which achieves a better average score in the arithmetic expression task. All reported values are evaluated by averaging across 5 BO trials.

Method	Expressions			Molecules		
	1st(↓)	2nd(↓)	3rd(↓)	1st(↑)	2nd(↑)	3rd(↑)
GVAE	0.10	0.46	0.52	3.13	3.10	2.37
CVAE	0.45	0.48	0.61	2.75	0.82	0.63
GCVVAE	0.39	0.40	0.43	3.22	2.83	2.63
GRAE	0.39	**0.39**	0.43	3.74	3.52	**3.14**
Ours	**0.03**	0.40	**0.41**	**4.15**	**3.84**	3.12

Table 4.3 Best scores found by each method for arithmetic expression and molecule experiments. Baseline values reported from [60].

Method	Expressions		Molecules	
	Frac. valid (↑)	Avg. score (↓)	Frac. valid (↑)	Avg. score (↑)
GVAE	0.99 ± 0.01	3.26 ± 0.20	0.28 ± 0.04	-7.89 ± 1.90
CVAE	0.82 ± 0.07	4.74 ± 0.25	0.16 ± 0.04	-25.64 ± 6.35
GCVVAE	0.99 ± 0.01	**2.85 ± 0.08**	0.76 ± 0.06	-6.40 ± 0.80
GRAE	**1.00 ± 0.00**	3.22 ± 0.03	0.72 ± 0.09	-5.62 ± 0.71
Ours	**1.00 ± 0.00**	3.32 ± 0.04	**0.72 ± 0.03**	**-5.08 ± 1.30**

Table 4.4 Fraction of valid samples and their corresponding average scores for arithmetic expression and molecule experiments for each method. Baseline values reported from [60].

Objective	Method	Expressions	Molecules
LL	GVAE	-1.320 ± 0.001	-1.739 ± 0.004
	CVAE	-1.397 ± 0.003	-1.812 ± 0.004
	Ours	**-1.309 ± 0.001**	**-1.689 ± 0.003**
RMSE	GVAE	0.884 ± 0.002	1.404 ± 0.006
	CVAE	0.975 ± 0.004	1.504 ± 0.006
	Ours	**0.877 ± 0.001**	**1.400 ± 0.002**

Table 4.5 Predictive performances of sparse Gaussian processes on different VAEs. Baseline values are taken from [114].

Predictive Performance of the Latent Representation Similar to [114], we also evaluate the predictive performance of the latent representations of the proposed model. The sparse Gaussian process model used in the BO evaluates the predictive performance on a left out 10% of data (test). The input to the sparse GP model is the test data (formed by the latent representation of the available sequences), and the output is the prediction of each task's associated properties/scores. The test log-likelihood and the average RMSE values obtained for our model are compared to GVAE [114] and CVAE [73] in Table 4.5. Our model yields better predictive performance on both tasks, showing that the proposed model learned better latent features for better predictions than the other two baseline models.

4.4.5 Ablation Study

In this section, we perform an ablation study on the two loss terms in the proposed regularization loss.

Quantitative analysis. We consider the MNIST dataset for quantitative evaluation of the ablation study on the regularization loss terms in the proposed model. When the model is trained without the KS distance loss for MNIST images, we observed an FID of 49.82, and when trained without the covariance matching loss, we observed an FID of 38.45. These values are significantly worse than the FID we achieve when training with the weighted combination of both regularization losses i.e 13.11. These empirical evaluations show that combining the two regularization terms facilitates a better prior-posterior match and hence better image generation.

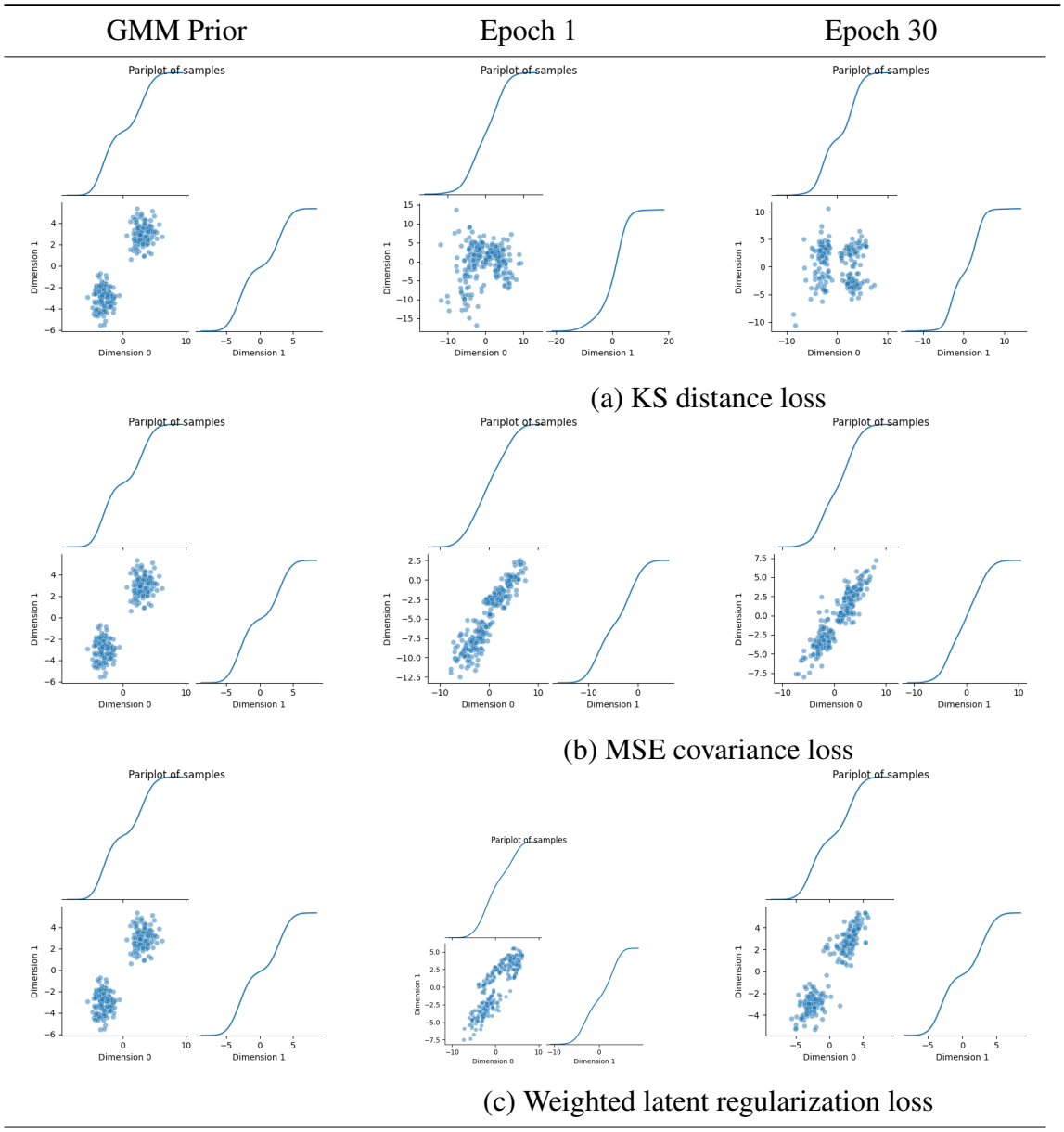


Fig. 4.9 Ablation study on loss functions - 2D pair plot visualization of the target prior and posterior (test images) of the proposed model trained on a subset of MNIST images with different terms of the loss functions.

Visual analysis. For simplicity, we consider a subset of MNIST images with two digits 1 and 8 as the training dataset for this line of experiments. For ease of visualization, the prior is chosen to be a mixture of two Gaussian with means $(3,3)$ and $(-3,-3)$ and identity covariance matrices. We train a deterministic autoencoder with the two individual loss terms (mean squared covariance distances and simplified Kolmogorov-Smirnov distance) and the proposed weighted combination of both. In Figure 4.9, we show the latent representations of the training data using these three regularizers after 1 and 30 epochs. It can be seen that a combination of the proposed two loss terms is essential for effectively regularizing the latent representations to match the target prior.

4.4.6 Hyperparameter Sensitivity Analysis

From a conceptual point of view, the essential hyperparameters of our model are (a) the weights of the different terms in the training objective and (b) the number of components in the prior. Section 3.4 proposes an explicit way to fix the loss function's weights. We investigate the sensitivity of our model performance to the number of components in the GMM prior. We trained our model on the MNIST dataset using a GMM prior with $1, 5, 10, 15$, 20, and 25 modes, respectively. The observed FID scores for the respective number of components are shown in Figure 4.10. The result shows that with an increasing number of components in the chosen prior, the performance of our model improves significantly. Consequently, choosing a large number of components can benefit practical considerations.

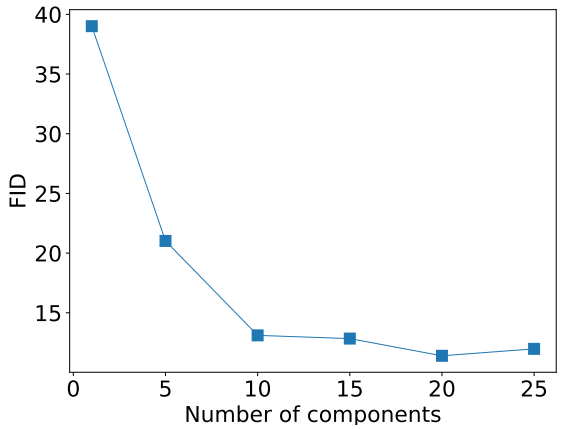


Fig. 4.10 Hyperparameter sensitivity analysis - FID of the MNIST generated samples when the model is trained with a different number of components in the GMM prior.

4.4.7 Network Architecture and Implementation Details

Image generation. The architectural details of the encoder and decoder used are shown in Table 4.6. For a fair comparison, we used the same architecture for all the baseline methods. Filter size of 4 is used for all layers in the network, with padding size 1 and stride 2. Please refer to the code appendix for the implementation of the proposed model. For regularized autoencoders (RAE) and other VAEs in the baseline comparison, we used the official GitHub repository [60] to evaluate the results.

Dataset	Encoder		Decoder	
	Layer	Output	Layer	Output
MNIST/	Input	$1 \times 32 \times 32$	Input	10×1
FASHION MNIST	Conv2D, BN, ReLU	$128 \times 16 \times 16$	FC, Reshape	$1024 \times 2 \times 2$
	Conv2D, BN, ReLU	$256 \times 8 \times 8$	Conv2DT, BN, ReLU	$512 \times 4 \times 4$
	Conv2D, BN, ReLU	$512 \times 4 \times 4$	Conv2DT, BN, ReLU	$256 \times 8 \times 8$
	Conv2D, BN, ReLU	$1024 \times 2 \times 2$	Conv2DT, BN, ReLU	$128 \times 16 \times 16$
	Flatten, FC	10×1	Conv2DT, BN, ReLU	$1 \times 32 \times 32$
SVHN	Input	$3 \times 32 \times 32$	Input	100×1
	Conv2D, BN, ReLU	$128 \times 16 \times 16$	FC, Reshape	$1024 \times 2 \times 2$
	Conv2D, BN, ReLU	$256 \times 8 \times 8$	Conv2DT, BN, ReLU	$512 \times 4 \times 4$
	Conv2D, BN, ReLU	$512 \times 4 \times 4$	Conv2DT, BN, ReLU	$256 \times 8 \times 8$
	Conv2D, BN, ReLU	$1024 \times 2 \times 2$	Conv2DT, BN, ReLU	$128 \times 16 \times 16$
	Flatten, FC	100×1	Conv2DT, BN, ReLU	$3 \times 32 \times 32$
CELEBA	Input	$3 \times 64 \times 64$	Input	64×1
	Conv2D, BN, ReLU	$128 \times 32 \times 32$	FC, Reshape	$1024 \times 4 \times 4$
	Conv2D, BN, ReLU	$256 \times 16 \times 16$	Conv2DT, BN, ReLU	$512 \times 8 \times 8$
	Conv2D, BN, ReLU	$512 \times 8 \times 8$	Conv2DT, BN, ReLU	$256 \times 16 \times 16$
	Conv2D, BN, ReLU	$1024 \times 4 \times 4$	Conv2DT, BN, ReLU	$128 \times 32 \times 32$
	Flatten, FC	64×1	Conv2DT, BN, ReLU	$3 \times 64 \times 64$

Table 4.6 Encoder and Decoder network architecture - Image generation. Conv2D stands for the convolution layer, BN corresponds to batch normalization, Conv2DT refers to the transposed convolution layer, and FC stands for the fully connected layer.

We train our model with ADAM optimizer [105], using a batch size of 100, the number of epochs 100, momentum (β_1, β_2) = (0.5, 0.999) with starting learning rate of 0.002 which exponentially decays when the validation loss plateaus. MNIST and FASHION-MNIST's latent space dimensions are 10, 100 for SVHN, and 64 for CELEBA images. The image reconstruction loss coefficient value of 0.005 is used for all experiments. The other two loss coefficient values can be calculated as mentioned in section 3 of the main paper. For the prior definition, we define the means of each Gaussian component as one hot encoding vector

with a standard deviation of 1. A mixture of 10 components with equally weighted mixing coefficients was used for MNIST, FASHION MNIST, and SVHN, and 20 for CELEBA images. For evaluation metrics, Fréchet Inception Distance (FID) [78] is calculated for 10000 images and averaged across five different runs. The FIDs observed by sampling from the prior along with error bars (for different runs) are as follows, MNIST: 13.11 ± 0.9, FASHION-MNIST: 33.70 ± 0.8, SVHN: 37.42 ± 1.1 and CELEBA: 49.79 ± 1.2. And the FIDs that we observe after fitting a GMM to the latent space of our model are as follows (for different runs), MNIST: 12.82 ± 0.6, FASHION-MNIST: 26.62 ± 0.8, SVHN: 36.46 ± 0.9 and CELEBA: 44.79 ± 1.0. All our experiments were conducted on a single GTX1080 GPU with 12/16 GB RAM. Since the cluster is part of a carbon-neutral framework, these experiments did not contribute to climate change.

Modeling discrete data. We extend the official Tensorflow implementation of GRAMMAR-VAE [114] with our novel regularizer to evaluate the experiments. The image reconstruction loss coefficient used is 0.005 for both experiments. The other two-loss coefficient values can be calculated as mentioned in section 4.3.3 of the leading paper. We used the same network architecture and other hyper-parameters similar to the original implementation.

For the *arithmetic expression fitting* task, the model is trained with a dataset of $100,000$ randomly generated univariate arithmetic expressions (functions of x) following a defined grammar [114]. The objective of this experiment is to search in the latent space of the trained model to find an expression that best matches a fixed target dataset. The target dataset is defined by selecting 1000 input values of x, which is linearly spaced between *minus*10 and 10. The corresponding x values are given to the true function $1/3 + x + \sin(x * x)$ to generate target observations. The target variable/score to optimize is defined as the $\log(1 +$ MSE) between the predictions made by an expression and the true data.

For the *molecule discovery task,* the model is trained with a dataset of $250,000$ SMILES strings ZINC250K [73] following the context-free grammar as defined in [114]. The latent space of the trained model is then traversed to find the molecule with the best drug-likeliness score. The molecule's design metric water octanol partition coefficient (logP) quantifies the drug likeliness score.

4.5 Conclusion

Recent studies have illustrated the effectiveness of flexible priors in VAEs to learn more meaningful latent representations. Following recent work highlighting the potential of deterministic alternatives to the variational formulation in VAEs, we propose a simple

deterministic autoencoding framework with more powerful regularizers to accommodate expressive multi-modal priors. Our experimental evaluations show that the proposed training objective yields comparable sampling quality to those of variational autoencoders and achieves better performance in modeling complex discrete data structures.

Chapter 5

Towards Robust Deterministic Autoencoders

The susceptibility of Variational Autoencoders (VAEs) to adversarial attacks indicates the necessity to evaluate the robustness of the learned representations along with the generation performance. The vulnerability of VAEs has been attributed to the limitations associated with their variational formulation. Deterministic autoencoders could overcome the practical limitations associated with VAEs and offer a promising alternative for image generation applications. In this chapter, we propose an adversarially robust deterministic autoencoder with superior performance in both generation and robustness of the learned representations. We introduce a regularization scheme to incorporate adversarially perturbed data points to the training pipeline without increasing the computational complexity or compromising the generation fidelity compared to the robust VAEs by leveraging a loss based on the two-point Kolmogorov–Smirnov test between representations. We conduct extensive experimental studies on popular image benchmark datasets to quantify the robustness of the proposed approach based on the adversarial attacks targeted at VAEs. Our empirical findings show that the proposed method achieves significant performance in both robustness and fidelity when compared to the robust VAE models. This work is published at the Conference on Neural Information Processing Systems (NeurIPS), 2022 [162].

5.1 Introduction

One of the significant advantages of Variational autoencoders (VAEs) is that they provide semantically meaningful latent representations of high-dimensional complex input distributions, which can be further utilized for various downstream tasks [80, 74, 165, 59]. However,

as discussed in Chapter 4, VAEs are often limited due to theoretical and practical limitations such as over-regularization and prior-posterior mismatch resulting in trade-offs between generation and reconstruction fidelity [179, 34, 60]. The regularized deterministic autoencoder introduced in the previous chapter offers promising alternatives to overcome these limitations.

The semantically meaningful representations learned by VAEs can still be corrupted by so-called adversarial attacks [174, 110, 115], where even small but specifically crafted changes to the input can lead to very different reconstructions. This observation reveals a lack of generalization within such models and is, therefore, a serious concern with respect to many practical applications. While it is harder to attack VAEs when compared to classifier networks [65] it is essential to analyze the robustness of VAEs along with their generative performance, to validate whether the learned latent representations are meaningful. Hence, there has been increasing research interest in the deep learning field toward training robust models for classifiers and autoencoders, i.e., for robust representation learning. Borrowed from the training of robust classification models [125, 196], the concept of adversarial training has proven to be able to smoothen the VAE encoder and improve the robustness of the learned representations [23]. Other attempts toward learning robust representation spaces introduce either complex network architectures or expensive regularization mechanisms to improve the robustness of VAEs [193, 11]. Further, previous works have pointed out that the robustness of VAEs can be improved by generating disentangled latent representations or by encouraging the smoothness or consistency of the encoding-decoding process [193, 22]. However, regularizing the VAE objective to enhance robustness often leads to poor generation ability compared to its non-robust counterpart. Hence, we seek to focus on improving the robustness of autoencoders while still maintaining the generation performance.

In this chapter, we introduce a simple and easy to train deterministic autoencoder that exhibits superior performance in generation and adversarial robustness. We argue that the deterministic approach enhances the robustness of VAEs when the latent codes are correctly regularized. Consequently, we extend the training objective of the multi-modal deterministic autoencoders introduced in the last chapter to incorporate adversarially perturbed input data points in the latent space. We conduct extensive experimental analysis to evaluate the robustness of the trained model on popular benchmarks such as MNIST, FASHIONMNIST, SVHN, and CELEBA images. Our empirical evaluations show that the proposed model consistently exhibits high adversarial robustness and significantly better generation performance than state-of-the-art robust VAE baseline models. We also show that by improving the robustness of the learned representations, a classifier trained on the learned latent space of the model also exhibits better robustness.

5.2 Related Work

The ability to defend against adversarial attacks is closely related to the sensitivity of the learned latent representations to slight changes in the input data points. In this section, we review adversaries for VAEs and the strategies proposed to defend against adversarial attacks.

Adversarial attacks on VAEs. Adversarial attacks targeted towards autoencoders were first discussed in [65]. Common attacks on VAEs follow procedures similar to attacks against classifiers, i.e., they aim to maximize the network's loss. Usually, slight perturbations are added to the input images to make the reconstructions similar to a specific target image (targeted/supervised attack) or a completely different image (untargeted/unsupervised) [174, 110] such as to maximize the reconstruction loss. In [193] Willets et al. show that applying the TC regularization introduced in TC-VAEs to hierarchical VAEs yields robust representations. Although the resulting model improves adversarial robustness, the training complexity is high compared to a VAE. Cemgil et al. [23] relate the robustness of VAEs to the smoothness of the encoding process. Similarly to Madry et al. [125], they introduce a regularization scheme based on a selection mechanism in the latent space to generate additional data points to minimize the entropy regularized Wasserstein distance between latent representations. Following the same direction, Cemgil et al. [22] argue that the lack of consistency between the encoding-decoding process causes the susceptibility to adversarial attacks in VAES. In contrast to the previous works, Camuto et al. [20] provide a theoretical insight into the robustness of VAEs and introduce a novel criterion for robustness in VAEs. Barret et al. [11] propose constraining the Lipschitz constants for both encoder and decoder to ensure certifiable adversarial robustness of VAEs. Although these methods improve the adversarial robustness of VAEs, they are often accompanied by complex network architectures and expensive training procedures. In contrast, our approach adopts an inexpensive adversarial training scheme for the latent space of deterministic autoencoders by an elegant extension to the regularization proposed in the last chapter to ensure robustness and fidelity.

Adversarial training. Adversarial training is one of the most straightforward and intuitive methods to train robust models. Many research works have been proposed in this direction for classifier-based networks. The basic idea is to create adversarial examples and incorporate them into the model's training process. The robustness achieved by employing adversarial training highly depends on the type and strength of the adversarial examples used in training [196]. This section summarizes some well-known and widely used techniques for training adversarially robust classifier models, which also inspired the adversarial training scheme in the proposed method for VAEs.

Followed by the discovery of adversarial examples [173], the Fast Gradient Sign Method (FGSM) was introduced to generate adversarial samples by a single gradient step [68]. FGSM computes the gradient of the loss function with respect to the input sample and then considers the sign of the gradient to generate an adversarial sample that maximizes the loss. This method was further improved by a randomization step known as R+FGSM [182]. In subsequent research, the Basic Iterative Method [113] improved FGSM performance by applying several smaller FGSM steps, which were later improved by adding multiple random restarts. In [44], Dong et al. incorporated momentum into the iterative FGSM to stabilize the gradient update directions. The Projected Gradient Descent (PGD) [126] approach is similar to the iterative FGSM method, except that the initialization step is set to a random point in the L_p ball of interest around the sample and random restarts are performed. Since the introduction of PGD, several adversarial training methods have been proposed to improve robustness [203, 136, 181]. Despite the advances in this field, PGD-based adversarial training is still considered an effective and reliable approach for developing robust models [169, 196]. However, running a strong PGD adversary during training is very expensive. In [196], Wong et al. show that FGSM-based adversarial training combined with random initialization achieves similar performance to PGD-based training. Wong et al. also identify several failure modes that could lead to FGSM failure and present several tricks and techniques to further improve the potential of FGSM-based training. Fast-FGSM allows for much cheaper adversarial training while being as effective as its expensive PGD counterpart. We adapt Fast-FGSM to the latent space of the proposed model to generate adversarial samples in Section 5.3.2.

5.3 Adversarially Robust Deterministic Autoencoder

Deterministic Autoencoders offer a promising alternative to VAEs for learning meaningful representations of complex input spaces with high fidelity. Motivated by this fact, we aim to explore further the robustness of the learned representations of the resulting model. We are particularly interested in the multimodal prior setting since a flexible and expressive Gaussian mixture model (GMM) prior assumption facilitates encoding similar data points closer together while distancing dissimilar points far in the latent space - a behavior that has also been found to be beneficial in learning robust classifier models [59]. Consequently, we propose to adopt the regularization technique proposed in Chapter 4 to regularize the learned latent representations of our model towards a predefined GMM prior. For a model to be inherently robust, slight perturbations in the input space should not result in substantial variations in the encoding space and the corresponding reconstructions. This could also be attributed to the

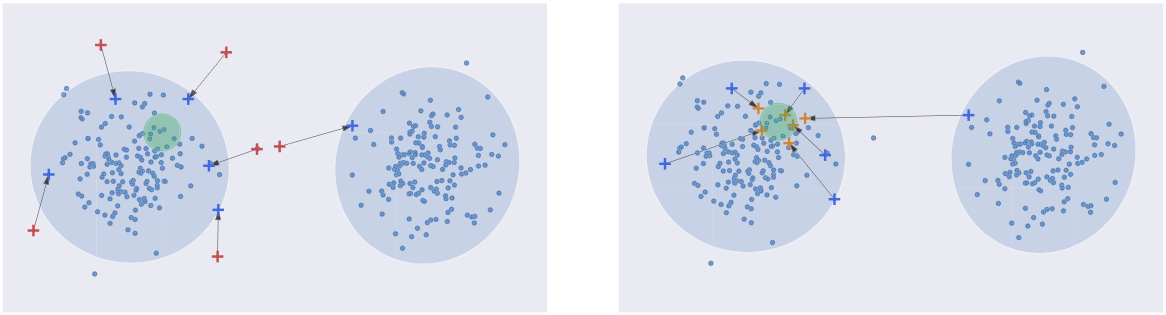


Fig. 5.1 Left: Learned latent representations in a deterministic autoencoder regularized towards a GMM prior with two components (blue-shaded regions). Consider a set of latent points $\mathbf{z}_1, \ldots, \mathbf{z}_N$ (blue dots) in a subspace (green shaded region) within a component and the corresponding adversarial examples $\mathbf{z}_1^{\text{adv}}, \ldots, \mathbf{z}_N^{\text{adv}}$ (red crosses). The adversarial examples tend to explore regions not covered by the input samples. If we assume that $\mathbf{z}$ and $\mathbf{z}^{\text{adv}}$ follow the same prior assumptions independently, the adversarial examples tend to move closer to the original samples (blue crosses). In the worst case scenario, an adversarial example might reside in a different component. Right: By establishing a strong coupling via a 2-point KS-distance regularization, the adversarial examples tend to move closer to the original samples (orange crosses) after regularization.

smoothness of the learned encoder. Hence, it is essential to investigate the smoothness of the learned representations and the reconstructions of the model. In the following subsections, we resume the regularization loss from the last chapter for completeness and illustrate how it can be extended to adversarial training samples.

5.3.1 Regularization of the learned representations

In the last chapter, the total loss of the model is a combination of two regularization terms as described in Section 4.3.2 to enforce the latent representations of the encoded data to match a predefined multi-modal prior distribution and a reconstruction loss. Motivated by [59], for equidistantly chosen modes μ_i as in Figure 5.1, we expect such a model to provide already not only an improved generation fidelity as observed in the previous chapter but also a more robust behavior. We provide empirical evidence for this conjecture in an ablation study in Section 5.4.4. The GMM regularization implicitly clusters latent points such that similar points are close to one another while dissimilar points are distant.

To further improve the robustness of our model, we employ adversarial training - a widely popular defense strategy utilized to learn robust deep networks [125, 196]. That is, during training, we utilize adversarial samples $\mathbf{z}_n^{\text{adv}}$ that are explicitly optimized to fall in an $\varepsilon-$ ball around the original latent representations $\mathbf{z}_n$ for $n \leq N$ and to decode to damaged or semantically altered images. In [4], it was observed that such adversarial samples tend

to explore the underrepresented regions in the latent space. In the following, we extend the losses in (4.7) and (4.4) to overcome this undesired behavior. This approach allows for cheaper yet very effective adversarial training while preserving the reconstruction and generation ability of the original model.

5.3.2 Adversarial Training Data Augmentation

To generate adversarial inputs, we adapt the fast gradient sign method [196] to the latent space of the model. For a given $\varepsilon > 0$ and datapoint $\mathbf{x}_n$, the objective of the attack is to find the corresponding adversary $\mathbf{x}_n^{\text{adv}}$ that would introduce maximum distance in the encoding space. That is, $\mathbf{x}_n^{\text{adv}} = \mathbf{x}_n + \delta_{\mathbf{x}_n}$, where $\delta_{\mathbf{x}_n}$ is the solution to the optimization problem

$$\arg\max_{\delta} \|g(\mathbf{x}_n + \delta) - g(\mathbf{x}_n)\|_2 \text{ s.t. } \|\delta\|_\infty \leq \varepsilon, \tag{5.1}$$

where g is the encoder of the model. To prevent adversarial samples from exploring unexplored regions of the latent space, we assume that the joint distribution of latent encodings $(\mathbf{z}_n, \mathbf{z}_n^{\text{adv}})$ (here $\mathbf{z}_n = g(x_n)$ and $\mathbf{z}_n^{\text{adv}} = g(x_n + \delta_{x_n})$) of data points and their adversarial samples $(\mathbf{x}_n, \mathbf{x}_n^{\text{adv}})$ to follow the same multi-modal GMM prior (Figure 5.1). One possible straightforward extension of the approach would be to consider adversarial examples as a specific data augmentation and regularize $\mathbf{z}_{1,\dots,N}$ and $\mathbf{z}_{1,\dots,N}^{\text{adv}}$ to the same GMM prior independently and ignoring cross-covariance between $\mathbf{z}_{1,\dots,N}$ and $\mathbf{z}_{1,\dots,N}^{\text{adv}}$ (here the off-diagonal elements in the covariance matrix of the GMM prior, that is the last matrix mentioned in equation (4.8), are zero). The corresponding losses in eqs. (4.7) and (4.8) take the form

$$\mathcal{L}_{\text{KS},k}^{\text{aug}}(\mathbf{z}_{1,\dots,N}, \mathbf{z}_{1,\dots,N}^{\text{adv}}) = \frac{1}{2}\mathcal{L}_{\text{KS},k}(\mathbf{z}_{1,\dots,N}) + \frac{1}{2}\mathcal{L}_{\text{KS},k}(\mathbf{z}_{1,\dots,N}^{\text{adv}}) \tag{5.2}$$

and

$$\mathcal{L}_{\text{CV},k}^{\text{aug}}(\mathbf{z}_{1,\dots,N}, \mathbf{z}_{1,\dots,N}^{\text{adv}}) = \frac{1}{4D^2}\sum_{\ell,d=1}^{2D}\left(\begin{bmatrix}\bar{\Sigma} & \bar{\Sigma}^{\text{cross}} \\ \bar{\Sigma}^{\text{cross}} & \bar{\Sigma}^{\text{adv}}\end{bmatrix}_{\ell,d} - \begin{bmatrix}\Sigma^{\text{GMM}} & 0 \\ 0 & \Sigma^{\text{GMM}}\end{bmatrix}_{\ell,d}\right)^2, \tag{5.3}$$

where $\bar{\Sigma}$ and $\bar{\Sigma}^{\text{adv}}$ are the empirical covariance matrices of the latent representations $\mathbf{z}_{1,\dots,N}$ and their adversaries $\mathbf{z}_{1,\dots,N}^{\text{adv}}$ respectively, and $\bar{\Sigma}_{\text{cross}}$ is the empirical cross-covariance between benign and adversarial samples. While such a regularization preserves the overall distribution even under adversarial attacks, it can not control the distance of a specific adversarial sample to its benign $\mathbf{z}_n$. In the worst case scenario, an adversarial example $\mathbf{z}_n^{\text{adv}}$ can be mapped to a

different Gaussian mixture component than $\mathbf{z}_n$ and therefore cause maximum damage in the reconstruction as shown in Figure 5.1(left).

To spread out the learned representations evenly, we inject Gaussian noise into the latent vectors during training. Let $\mathbf{x}_{\varepsilon,n}$ be the output of the decoder at $\mathbf{z}_n + \varepsilon_n$, where $\varepsilon_n \sim \mathcal{N}(0, I_D)$. The reconstruction loss equals the mean squared error between inputs $\mathbf{x}_n$ and their noisy reconstructions $\mathbf{x}_{\varepsilon,n}$.

5.3.3 A Two-Point KS-distance loss

To ensure that the adversarial examples remain in close proximity to the original mapping in the learned latent space, we establish a strong coupling between the two distributions, $\mathbf{z}_{1,\dots,N}$ and $\mathbf{z}^{\text{adv}}_{1,\dots,N}$. Hence we propose to match the empirical CDFs of $\mathbf{z}_{1,\dots,N}$ and $\mathbf{z}^{\text{adv}}_{1,\dots,N}$ and introduce a novel regularization based on the two-point KS-test [171]. By analogy to the one-point KS-test that tests whether a sample is drawn from a given, continuous distribution, the two-sample KS test determines whether two samples with empirical CDF are drawn from the same distribution. To this end, the two-sample KS test evaluates the supremum of the distance between the two CDFs. Here, we propose to minimize this distance computed from the marginalized two-point KS-test to align the distributions of benign points and their adversaries. The resulting loss is consistent with the previous regularization and establishes the desired coupling between the representations efficiently. The first regularization loss of the adversarially extended model with pairwise coupling takes the following form

$$\mathcal{L}^{\text{adv}}_{\text{KS},k}(\mathbf{z}_{1,\dots,N}, \mathbf{z}^{\text{adv}}_{1,\dots,N}) = \frac{2}{3}\mathcal{L}^{\text{aug}}_{\text{KS},k}(\mathbf{z}_{1,\dots,N}, \mathbf{z}^{\text{adv}}_{1,\dots,N}) + \frac{1}{3D}\sum_{d=1}^{D}\underset{n=1}{\overset{N}{\text{MSE}}}\left(\bar{F}_d([\mathbf{z}_n]_d), \bar{F}^{\text{adv}}_d([\mathbf{z}^{\text{adv}}_n]_d)\right) \tag{5.4}$$

where $\bar{F}_d, \bar{F}^{\text{adv}}_d$ are the empirical CDFs of $\mathbf{z}$ and $\mathbf{z}^{\text{adv}}$ respectively.

The correlations between the latent representations and their adversarial samples must be considered separately. The degree or strength of the coupling is controlled by a coupling parameter $\mid \alpha \mid \leq 1$, where $\alpha = 1$ indicates the border condition where $\mathbf{z} = \mathbf{z}^{\text{adv}}$. The covariance loss of the extended model becomes,

$$\mathcal{L}^{\text{adv}}_{\text{CV},k}(\mathbf{z}_{1,\dots,N}, \mathbf{z}^{\text{adv}}_{1,\dots,N}) = \frac{1}{4D^2}\sum_{\ell,d=1}^{2D}\left(\begin{bmatrix}\bar{\Sigma} & \bar{\Sigma}^{\text{cross}} \\ \bar{\Sigma}^{\text{cross}} & \bar{\Sigma}^{\text{adv}}\end{bmatrix}_{\ell,d} - \begin{bmatrix}\Sigma^{\text{GMM}} & \alpha\Sigma^{\text{GMM}} \\ \alpha\Sigma^{\text{GMM}} & \Sigma^{\text{GMM}}\end{bmatrix}_{\ell,d}\right)^2. \tag{5.5}$$

The total training objective of the model takes the following form,

$$\mathcal{L}(\mathbf{x}_{1,\dots,N}, \mathbf{x}^{\text{adv}}_{1,\dots,N}) = \lambda_{\text{REC}}\mathcal{L}_{\text{REC}}(\mathbf{x}^{\varepsilon}_{1,\dots,N}) + \lambda_{\text{KS}}\mathcal{L}^{\text{adv}}_{\text{KS},k}(\mathbf{z}_{1,\dots,N}, \mathbf{z}^{\text{adv}}_{1,\dots,N}) + \lambda_{\text{CV}}\mathcal{L}^{\text{adv}}_{\text{CV},k}(\mathbf{z}_{1,\dots,N}, \mathbf{z}^{\text{adv}}_{1,\dots,N}). \quad (5.6)$$

The weights $\lambda_{\text{REC}}, \lambda_{\text{KS}}$ and λ_{CV} can be calculated by taking the statistics of samples from the GMM prior as described in Section 4.3.3.

Two-Point KS-distance loss for unimodal prior. In this section, we formulate the proposed adversarial scheme for unimodal Gaussian $Z \sim \mathcal{N}(\mu, \Sigma)$ with mean μ and covariance matrix Σ. The first regularization loss of the adversarially extended model with pairwise coupling takes the following form,

$$\mathcal{L}^{\text{adv}}_{\text{KS},k}(\mathbf{z}_{1,\dots,N}, \mathbf{z}^{\text{adv}}_{1,\dots,N}) = \frac{2}{3}\mathcal{L}^{\text{aug}}_{\text{KS},k}(\mathbf{z}_{1,\dots,N}, \mathbf{z}^{\text{adv}}_{1,\dots,N}) + \frac{1}{3D}\sum_{d=1}^{D}\underset{n=1}{\overset{N}{\text{MSE}}}\left(\bar{F}_d([\mathbf{z}_n]_d), \bar{F}^{\text{adv}}_d([\mathbf{z}^{\text{adv}}_n]_d)\right) \quad (5.7)$$

where $\bar{F}_d, \bar{F}^{\text{adv}}_d$ are the empirical CDFs of $\mathbf{z}$ and $\mathbf{z}^{\text{adv}}$ respectively.

The correlations between the latent representations and their adversarial samples must be considered separately. The covariance loss of the extended model is defined as follows,

$$\mathcal{L}^{\text{adv}}_{\text{CV},k}(\mathbf{z}_{1,\dots,N}, \mathbf{z}^{\text{adv}}_{1,\dots,N}) = \frac{1}{2D^2}\sum_{\ell,d=1}^{2D}\left(\begin{bmatrix}\bar{\Sigma} & \bar{\Sigma}^{\text{cross}} \\ \bar{\Sigma}^{\text{cross}} & \bar{\Sigma}^{\text{adv}}\end{bmatrix}_{\ell,d} - \begin{bmatrix}\Sigma & \alpha\Sigma \\ \alpha\Sigma & \Sigma\end{bmatrix}_{\ell,d}\right)^2. \quad (5.8)$$

where $\bar{\Sigma}$ is the empirical covariance matrix of the latent representations, Σ stands for the prior covariance and $\alpha \leq 1$ is the coupling parameter. Since we consider a Gaussian prior with zero mean and identity covariance, $Z \sim \mathcal{N}(0, I)$, the covariance loss becomes,

$$\mathcal{L}^{\text{adv}}_{\text{CV},k}(\mathbf{z}_{1,\dots,N}, \mathbf{z}^{\text{adv}}_{1,\dots,N}) = \frac{1}{2D^2}\sum_{\ell,d=1}^{2D}\left(\begin{bmatrix}\bar{\Sigma} & \bar{\Sigma}^{\text{cross}} \\ \bar{\Sigma}^{\text{cross}} & \bar{\Sigma}^{\text{adv}}\end{bmatrix}_{\ell,d} - \begin{bmatrix}I & \alpha I \\ \alpha I & I\end{bmatrix}_{\ell,d}\right)^2. \quad (5.9)$$

The total training objective of the model takes the following form,

$$\mathcal{L}(\mathbf{x}_{1,\dots,N}, \mathbf{x}^{\text{adv}}_{1,\dots,N}) = \lambda_{\text{REC}}\mathcal{L}_{\text{REC}}(\mathbf{x}^{\varepsilon}_{1,\dots,N}) + \lambda_{\text{KS}}\mathcal{L}^{\text{adv}}_{\text{KS},k}(\mathbf{z}_{1,\dots,N}, \mathbf{z}^{\text{adv}}_{1,\dots,N}) + \lambda_{\text{CV}}\mathcal{L}^{\text{adv}}_{\text{CV},k}(\mathbf{z}_{1,\dots,N}, \mathbf{z}^{\text{adv}}_{1,\dots,N}). \quad (5.10)$$

The weights λ_{KS} and λ_{CV} can be calculated by taking the statistics of samples from the Gaussian prior as mentioned in Section 4.3.3.

5.4 Experiments and Results

We conduct an extensive experimental analysis to evaluate the robustness of the proposed model. We consider the state-of-the-art latent space attacks targeted at VAEs [65, 110, 11] and evaluate the robustness based on the quantitative metrics as described in Section 5.4.3. Since the latent spaces of VAEs are often further utilized for various downstream applications, we also consider the impact of such adversarial attacks on a classifier trained in the latent space of the model. To evaluate the fidelity of the learned representations, we report the Fréchet inception distance (FID) [78] of the generated images. Our model is compared with the following baseline models, Variational Autoencoder (VAE) [108], β-VAE [79], β-TCVAE [26], LipschitzVAE [11], Smooth Encoders (SE) [23] and Autoencoding Variational Autoencoder (AVAE) [22]. The experimental study is conducted on important image benchmark datasets such as MNIST, FASHIONMNIST, SVHN, and CELEBA. For simplicity, we consider a fully connected network architecture for experiments on MNIST and FASHIONMNIST images and a convolutional architecture for experiments on SVHN and CELEBA images.

5.4.1 Adversarial Attacks on Variational Autoencoders

Adversarial attacks targeted at VAEs attempt to add small noise perturbations to the input data points that fool the model into reconstructing the input image to a target adversarial image or a completely different image. Following recent literature [11, 193], we consider two types of adversarial attacks in our experiments.

Latent space attack. Latent space attacks or supervised attacks are considered the most effective mode of attack on VAEs. Here, the attacker tries to add a noise perturbation δ to a data point $\mathbf{x}$, such that the latent representation $\mathbf{z}_{x+\delta}$ of the perturbed input $\mathbf{x}+\delta$ is close to the latent representation $\mathbf{z}_t$ of a chosen target adversarial image, $\mathbf{x}_t$. The attack involves solving the following optimization problem,

$$\arg \min_{\|\delta\|_2 \leq \lambda} \|(\mathbf{z}_{x+\delta} - \mathbf{z}_t)\|_2. \tag{5.11}$$

Maximum damage attack. Further, we consider maximum damage or output space attack. In this setting, the adversary perturbs the input data point to cause maximum damage in the

reconstruction of the decoder f of the model and optimizes the following objective,

$$\arg \max_{\|\delta\|_2 \leq \lambda} \|f(\mathbf{z}_{x+\delta}) - f(\mathbf{z}_x)\|_2. \tag{5.12}$$

In both scenarios, the noise perturbation is explicitly constrained by some constant $\lambda > 0$ to ensure a consistent comparison with the baseline models.

5.4.2 Qualitative Analysis

For qualitative analysis, we provide visual results for MNIST, FASHIONMNIST, SVHN, and CELEBA images for both adversarial attacks - latent space and maximum damage attacks. For latent space attacks, we compare the source image, clean reconstruction, adversarial image, adversarial reconstruction, and target images across the dataset as shown in Figures 5.2 and 5.3. The adversarial reconstruction would strongly resemble the target image for a successful attack. As observed from the Figure, it can be seen that the proposed method remains more robust under latent space attacks when compared to the baseline models. For maximum damage attacks, we compare the source images, the corresponding clean reconstructions, adversarial images, and their corresponding reconstructions across the dataset as shown in Figures 5.4 and 5.5. These attacks are more successful when the adversarial reconstructions appear less similar to the clean reconstructions. As shown in the figures, both attacks get more successful with an increase in noise perturbation. However, for a given noise perturbation, the proposed method is more robust when compared to other models.

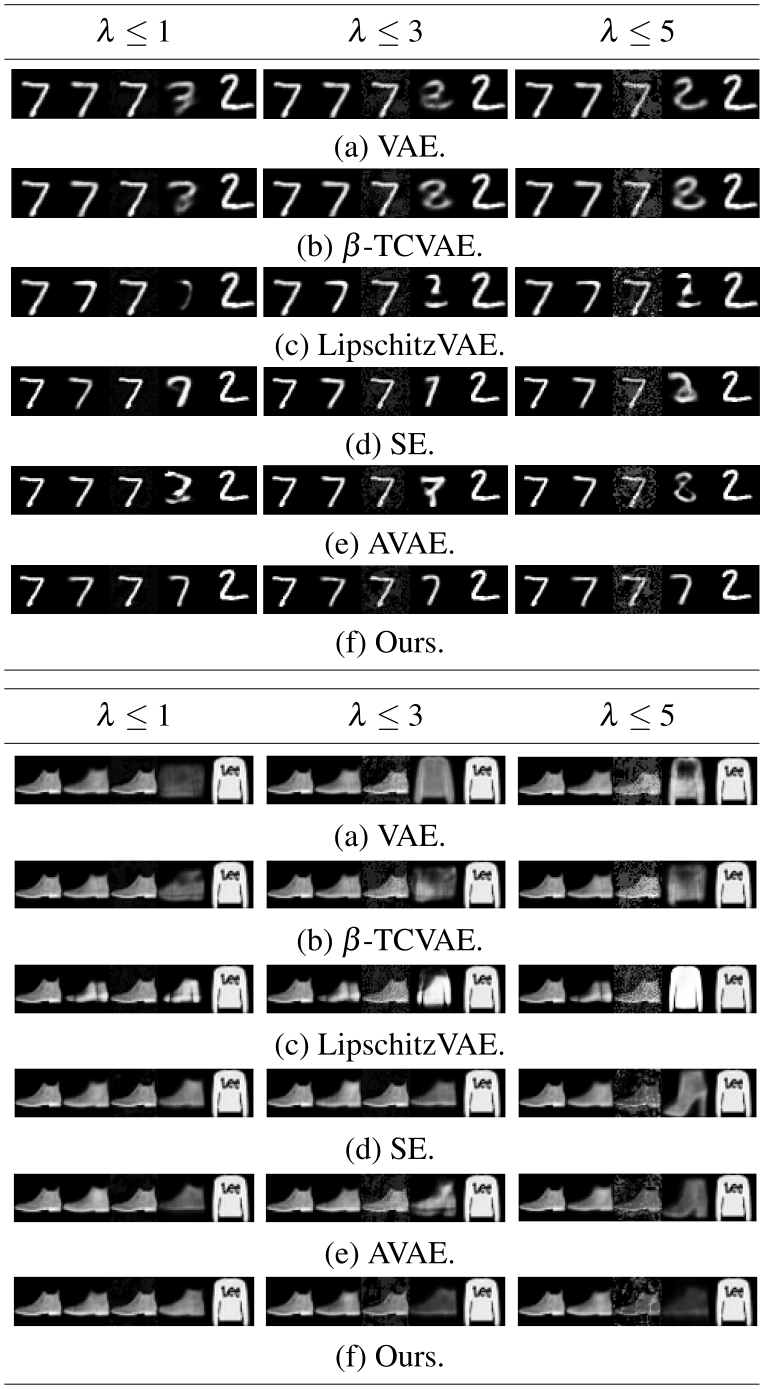


Fig. 5.2 Visual appraisal of latent space attacks on MNIST and FASHIONMNIST images with maximum input noise perturbation level λ limited to $1, 3$ and 5. (from left to right) Images in each row correspond to input image(x), clean reconstruction, adversarial image(x_a), adversarial reconstruction($\tilde{\mathbf{x}}_a$) and target image(x_t).

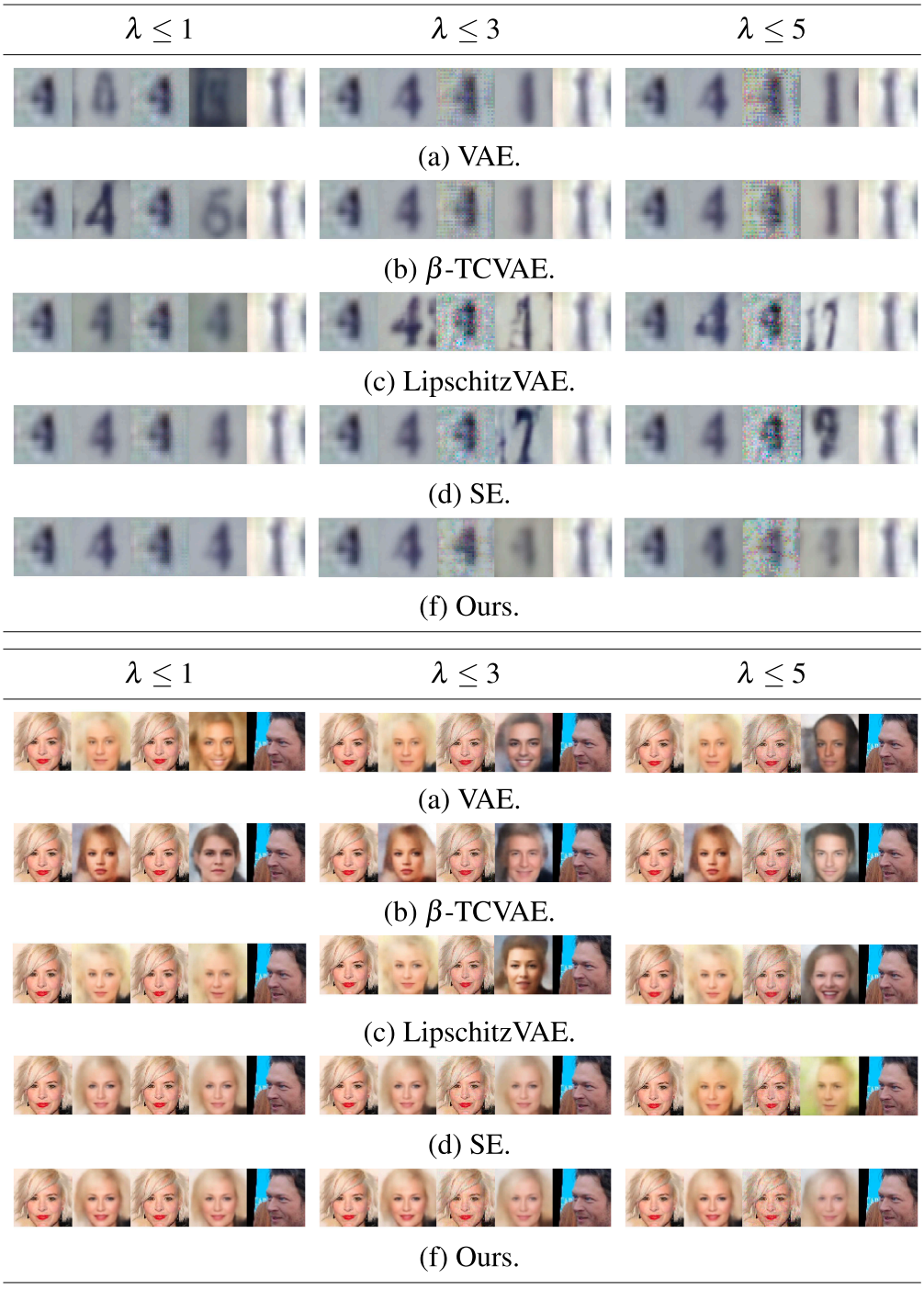


Fig. 5.3 Visual appraisal of latent space attacks on SVHN and CELEBA images with maximum input noise perturbation level λ limited to $1, 3$ and 5. (from left to right) Images in each row correspond to input image(x), clean reconstruction, adversarial image(x_a), adversarial reconstruction($\tilde{\mathbf{x}}_a$) and target image(x_t).

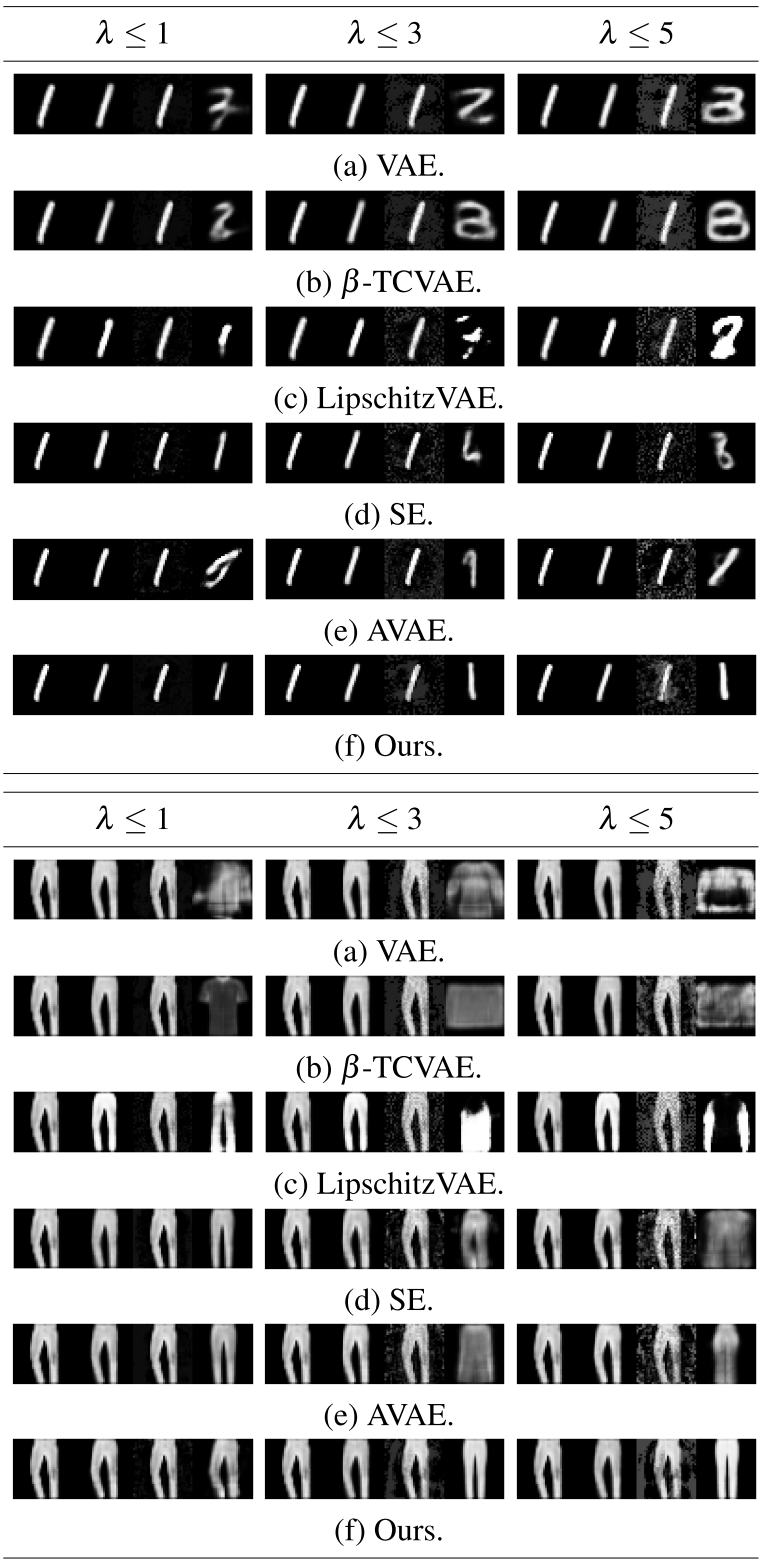


Fig. 5.4 Visual appraisal of maximum damage attacks on MNIST and FASHIONMNIST images with maximum input noise perturbation level λ limited to $1, 3$ and 5. (from left to right) Images in each row correspond to input image(x), clean reconstruction and adversarial image(x_a), adversarial reconstruction($\tilde{\mathbf{x}}_a$).

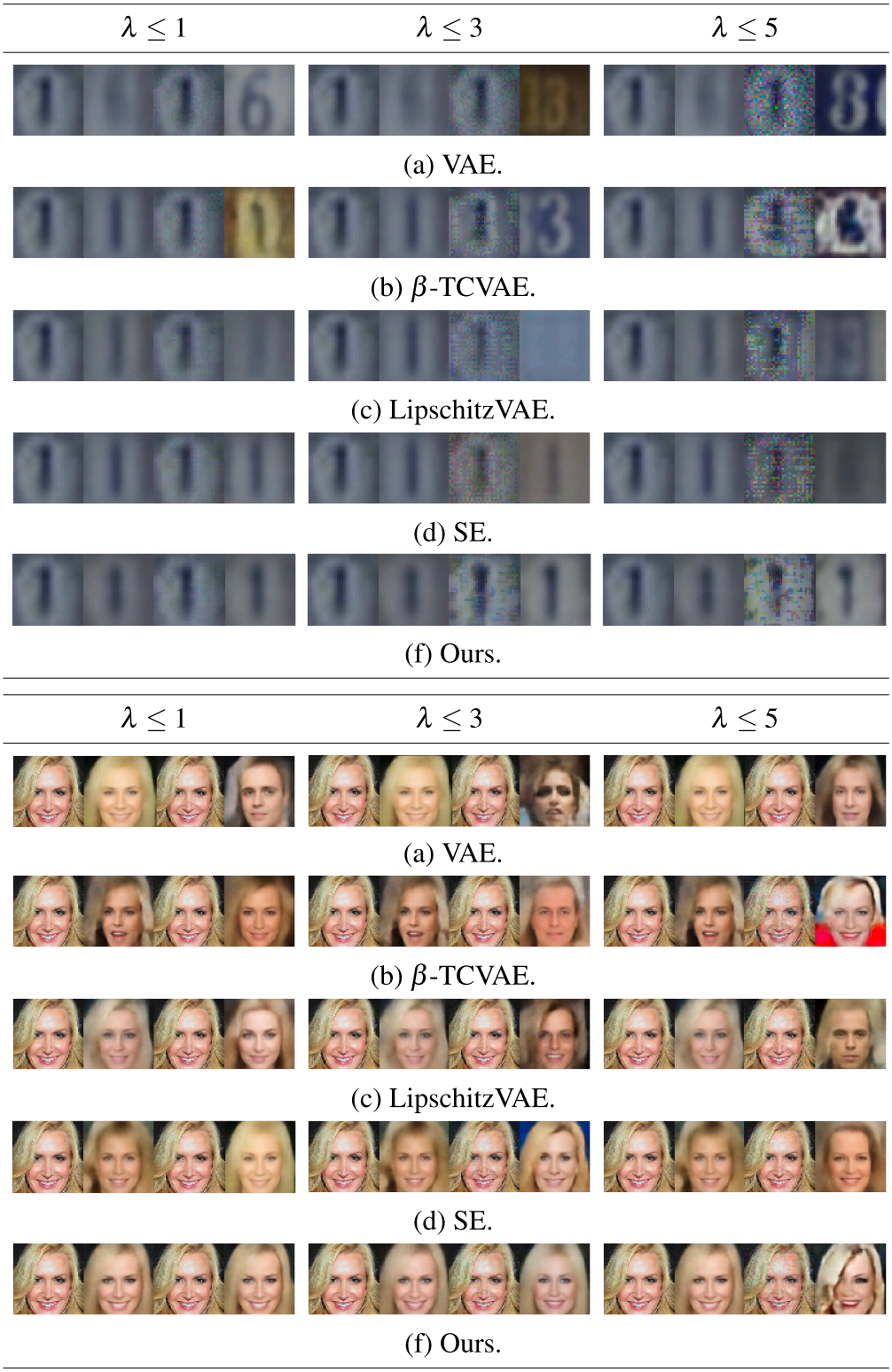


Fig. 5.5 Visual appraisal of maximum damage attacks on SVHN and CELEBA images with maximum input noise perturbation level λ limited to $1, 3$ and 5. (from left to right) Images in each row correspond to input image(x), clean reconstruction and adversarial image(x_a), adversarial reconstruction($\tilde{\mathbf{x}}_a$).

5.4.3 Quantitative Analysis

We consider three evaluation metrics to quantitatively estimate the robustness of the different models for the above-mentioned adversarial attacks.

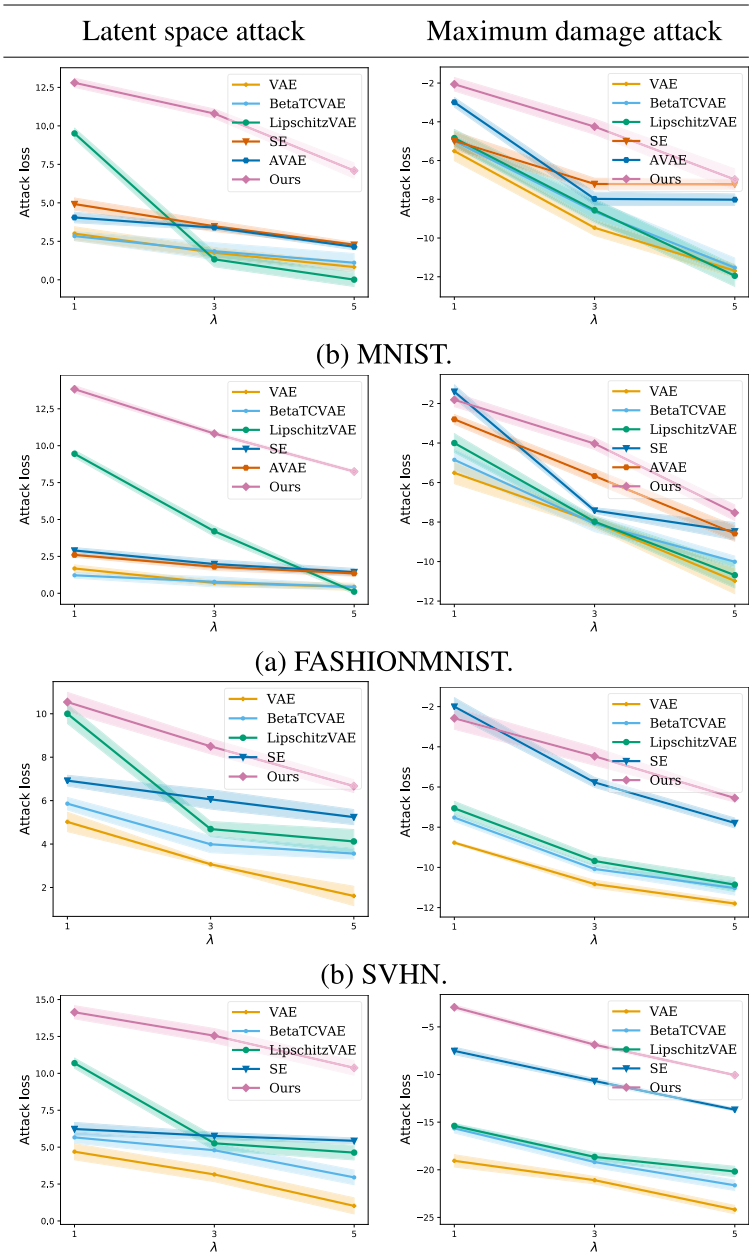
Attack loss. First, we evaluate the attack loss of the adversary. We report the achieved value of the optimization objectives in (5.11) and (5.12). The observed attack losses for the two forms of adversarial attacks are shown in Figure 5.6. Higher attack losses correspond to less successful attacks and hence better robustness. Due to the different regularization methods used in the baseline models, the inherent scale of the aggregated posterior changes and hence the value of the attack objective in eqn (5.11). Hence, the values reported in Figure 5.6 might not be directly comparable for latent space attacks.

Image similarity metric. Another strong indicator of the robustness of the considered models is the similarity between the images before and after the attack. Similar to [115], we use the perception-based similarity metric Multi-Scale Structural Similarity Index Measure (MSSSIM $\in [0,1]$) to compare the images. MS-SSIM is an advanced and more flexible version of the Structural Similarity Index Measure (SSIM). The intuition behind the SSIM metric is to utilize the structural information in the image to evaluate its quality. Since the human visual system can easily extract structural information from visuals, a measure of the same is considered an excellent approximation to access visual quality. Spatially adjacent pixels in the images have solid interdependencies and provide information regarding the structure of the objects in the images. SSIM is calculated between multiple patches on the two images to be compared. Consider two image patches, x, and y, at the same spatial location of the two images to be compared. Let μ_x, μ_y be the means, σ_x, σ_y be the variance of the x and y patch and σ_{xy} be the covariance between x and y, then the luminance (l), contrast (c) and structure (s) comparison are defined as follows,

$$l(x,y) = \frac{2\mu_x\mu_y C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{5.13}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{5.14}$$

$$s(x,y) = \frac{\sigma_{xy}\, C_3}{\sigma_x\sigma_y + C_3} \tag{5.15}$$



(b) MNIST.

(a) FASHIONMNIST.

(b) SVHN.

(a) CELEBA.

Fig. 5.6 Observed attack losses for latent space attack (eqn (5.11)) and maximum damage attack (eqn (5.12)) across dataset with varying λ values. We report the observed mean and standard deviation by attacking 100 randomly chosen test images in 10 different trials. Higher loss indicates more robustness.

where C_1, C_2, C_3 are small constants. The SSIM metric is defined as,

$$\text{SSIM}(x,y) = [l(x,y)].[c(x,y)].[s(x,y)] \tag{5.16}$$

MSSSIM, as the name suggests, is computed at multiple scales of images to be compared. The images to be compared are iteratively passed through multiple stages of a low-pass filter and downsampling to compute the contrast and structure measure described in equation 5.14 and 5.15. The luminance is calculated only at the highest scale, M. The MSSSIM is then calculated by combining the values at multiple scales as follows,

$$\text{MSSSIM}(x,y) = [l(x,y)]^{\alpha_M} \prod_{i=1}^{M} [c(x,y)]^{\beta_i}.[s(x,y)]^{\gamma_i} \tag{5.17}$$

where α_M, β, γ are parameters used to define the importance of each component.

To evaluate the robustness, we consider the similarity between the reference image $\mathbf{x}_r$ and the reconstructions $\tilde{\mathbf{x}_a}$ of the adversarially perturbed variants $\mathbf{x}_a$. In the case of latent space attacks, we consider the target image $\mathbf{x}_r = \mathbf{x}_t$ as the reference images and compare with the reconstruction $\tilde{\mathbf{x}_a} = f(\mathbf{z}_{\mathbf{x}+\delta})$ of the perturbed latent representation. For a maximum damage attack, we consider the original images $\mathbf{x}_r = \mathbf{x}$ as the reference images and compare them to $\tilde{\mathbf{x}_a}$. Lower values of MSSSIM$(x_r, \tilde{x_a})$ indicate less similarity between the reference (target) image and the adversarial reconstructions and correspond to less successful latent space attacks. And the higher value of MSSSIM$(\tilde{x_r}, \tilde{x_a})$ corresponds to less successful maximum damage attacks as they correspond to the high similarity between reference (original) and the adversarial reconstructions. We report the observed values in Table 5.1. Finally, it is also essential to evaluate how similar the adversarial images are to the original image. Ideally, a successful attack implies that both adversarial and original images look similar in appearance (MSSSIM ≈ 1). Hence we consider the MSSSIM between the original and adversarial images in Table 5.1.

We also report the l_2-distance as an alternative similarity metric to MSSSIM. To be precise, we compute the l_2-distance between the reference image $\mathbf{x}_r$ and the reconstructions $\tilde{\mathbf{x}_a}$ of the adversarially perturbed variants $\mathbf{x}_a$. Similar to the previous setup, under latent space attacks, we consider the target image $\mathbf{x}_r = \mathbf{x}_t$ as the reference images and report the $l2$ distance with the reconstruction $\tilde{\mathbf{x}_a} = f(\mathbf{z}_{\mathbf{x}+\delta})$ of the perturbed latent representation. For a maximum damage attack, we consider the original images $\mathbf{x}_r = \mathbf{x}$ as the reference images and the $l2$ distance with the reconstructions, $\tilde{\mathbf{x}_a}$. The observed values are given in Figure 5.7.

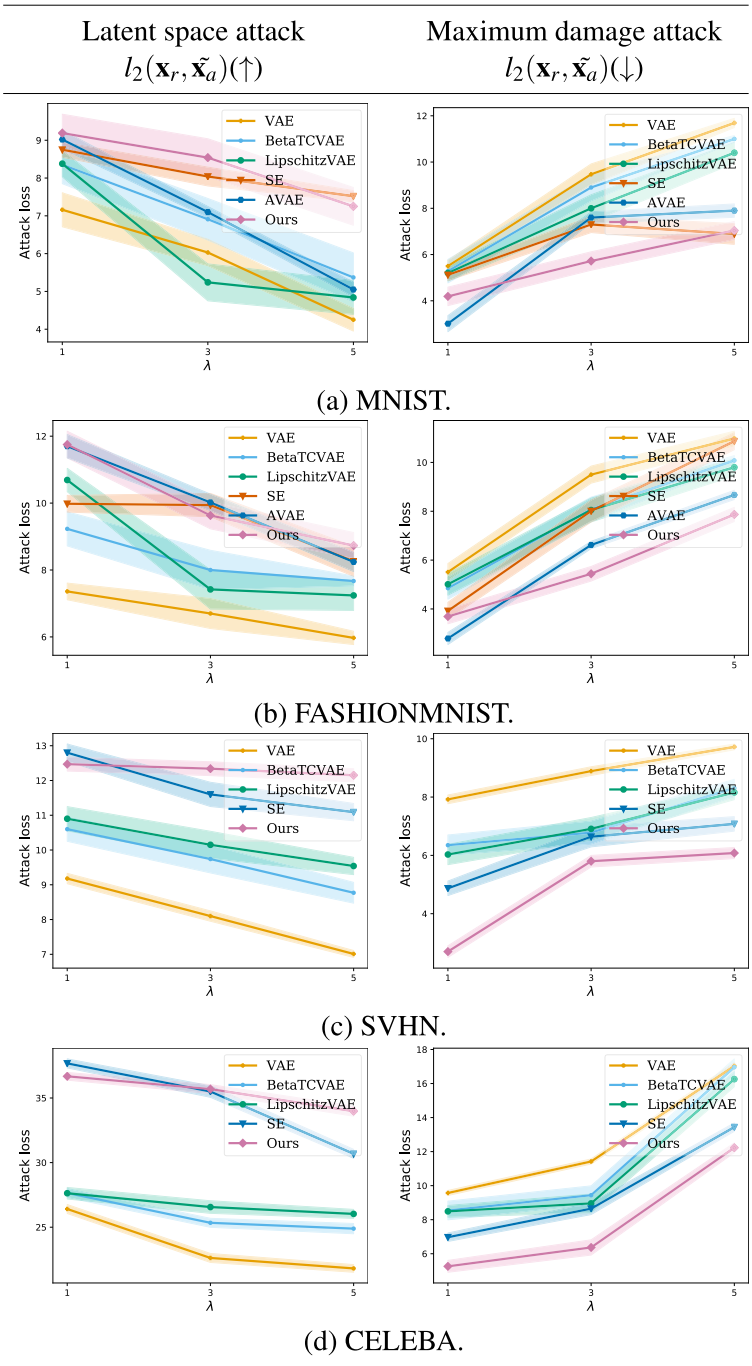
MNIST	Latent space attack						Maximum damage attack						FID(↓)
	MSSSIM($\mathbf{x}_r, \tilde{\mathbf{x}_a}$)(↓)			MSSSIM($\mathbf{x}_r, \mathbf{x}_a$)(↓)			MSSSIM($\tilde{\mathbf{x}_r}, \tilde{\mathbf{x}_a}$)(↑)			MSSSIM($\mathbf{x}_r, \mathbf{x}_a$)(↓)			
	1	3	5	1	3	5	1	3	5	1	3	5	
VAE	0.55	0.78	0.89	0.99	0.94	0.87	0.64	0.27	0.08	0.98	0.93	0.86	43.21
β-VAE	0.52	0.73	0.86	0.99	0.92	0.86	0.65	0.36	0.21	0.98	0.93	0.87	42.72
β-TCVAE	0.53	0.69	0.83	0.98	0.92	0.86	0.73	0.38	0.28	0.98	0.93	0.87	45.61
LipschitzVAE	0.50	0.68	0.79	0.98	0.93	0.89	0.75	0.41	0.33	0.98	0.93	0.86	59.45
SE	0.49	0.62	0.68	0.98	0.92	0.86	0.90	0.60	0.54	0.98	0.93	0.86	47.34
AVAE	0.49	0.59	0.62	0.98	0.91	0.83	0.80	0.65	0.59	**0.97**	0.89	0.86	48.47
Ours	**0.38**	**0.47**	**0.60**	**0.95**	**0.90**	**0.80**	**0.92**	**0.82**	**0.69**	0.98	**0.89**	**0.78**	**39.37**

FASHIONMNIST	Latent space attack						Maximum damage attack						FID(↓)
	MSSSIM($\mathbf{x}_r, \tilde{\mathbf{x}_a}$)(↓)			MSSSIM($\mathbf{x}_r, \mathbf{x}_a$)(↓)			MSSSIM($\tilde{\mathbf{x}_r}, \tilde{\mathbf{x}_a}$)(↑)			MSSSIM($\mathbf{x}_r, \mathbf{x}_a$)(↓)			
	1	3	5	1	3	5	1	3	5	1	3	5	
VAE	0.61	0.65	0.71	0.98	0.91	0.82	0.58	0.29	0.13	0.99	0.94	0.87	70.22
β-VAE	0.59	0.61	0.68	0.98	0.92	0.82	0.66	0.32	0.15	0.99	0.94	0.85	73.82
β-TCVAE	0.55	0.58	0.64	0.98	0.92	0.82	0.69	0.35	0.27	0.99	0.94	0.87	73.94
LipschitzVAE	0.43	0.59	0.67	0.99	0.94	0.89	0.71	0.34	0.30	0.99	0.94	0.88	79.45
SE	0.24	0.43	0.53	0.98	0.92	**0.81**	0.90	0.62	0.43	0.99	0.94	0.86	72.29
AVAE	0.32	0.34	**0.35**	0.98	0.92	0.82	0.79	0.52	0.45	0.99	0.94	0.92	74.45
Ours	**0.22**	**0.26**	0.39	**0.97**	**0.91**	**0.81**	**0.92**	**0.77**	**0.63**	**0.97**	**0.92**	**0.83**	**64.89**

SVHN	Latent space attack						Maximum damage attack						FID(↓)
	MSSSIM($\mathbf{x}_r, \tilde{\mathbf{x}_a}$)(↓)			MSSSIM($\mathbf{x}_r, \mathbf{x}_a$)(↓)			MSSSIM($\tilde{\mathbf{x}_r}, \tilde{\mathbf{x}_a}$)(↑)			MSSSIM($\mathbf{x}_r, \mathbf{x}_a$)(↓)			
	1	3	5	1	3	5	1	3	5	1	3	5	
VAE	0.46	0.76	0.87	0.98	0.87	0.77	0.55	0.46	0.38	0.99	0.93	0.89	58.98
β-VAE	0.44	0.70	0.81	0.99	0.89	0.77	0.52	0.49	0.47	0.99	0.93	0.88	61.65
β-TCVAE	0.39	0.65	0.72	0.99	0.89	0.77	0.63	0.60	0.54	0.99	**0.92**	0.88	62.59
LipschitzVAE	0.35	0.62	0.71	0.99	0.89	**0.76**	0.66	0.65	0.55	0.99	0.93	0.88	65.58
SE	0.19	0.33	0.34	0.99	0.92	0.81	0.79	0.69	0.60	0.99	0.96	0.94	61.28
Ours	**0.16**	**0.26**	**0.28**	**0.98**	**0.77**	**0.76**	**0.84**	**0.79**	**0.75**	**0.98**	**0.92**	**0.86**	**38.89**

CELEBA	Latent space attack						Maximum damage attack						FID(↓)
	MSSSIM($\mathbf{x}_r, \tilde{\mathbf{x}_a}$)(↓)			MSSSIM($\mathbf{x}_r, \mathbf{x}_a$)(↓)			MSSSIM($\tilde{\mathbf{x}_r}, \tilde{\mathbf{x}_a}$)(↑)			MSSSIM($\mathbf{x}_r, \mathbf{x}_a$)(↓)			
	1	3	5	1	3	5	1	3	5	1	3	5	
VAE	0.59	0.60	0.66	0.99	0.99	0.97	0.64	0.58	0.55	0.99	0.98	0.98	69.48
β-VAE	0.55	0.58	0.64	0.99	0.99	0.97	0.68	0.60	0.59	0.99	0.98	0.97	75.65
β-TCVAE	0.54	0.51	0.61	0.99	0.99	0.97	0.76	0.71	0.64	0.98	0.99	**0.96**	75.11
LipschitzVAE	0.49	0.51	0.55	0.99	**0.98**	0.97	0.73	0.70	0.64	0.98	**0.98**	**0.96**	77.89
SE	**0.27**	0.31	0.34	0.99	**0.98**	**0.96**	**0.97**	0.91	0.76	0.99	**0.98**	0.98	72.68
Ours	0.28	**0.29**	**0.32**	0.99	**0.98**	**0.96**	**0.97**	**0.93**	**0.80**	0.99	**0.98**	**0.96**	**51.98**

Table 5.1 Robustness evaluation across dataset - similarities between images in the event of latent space and maximum damage attacks in terms of MSSSIM. Here randomly chosen 100 test images are attacked in 10 different trials. x_r refers to reference image, x_a to adversarial image and $\tilde{x_r}$, $\tilde{x_a}$ to their corresponding reconstructions. The maximum input noise perturbation levels λ are limited to $1, 3$, and 5. Fidelity analysis - based on the FID of the generated images.



(a) MNIST.

(b) FASHIONMNIST.

(c) SVHN.

(d) CELEBA.

Fig. 5.7 Observed l_2 distance between images in the event of latent space and maximum damage attacks across datasets. Here, randomly chosen 100 test images are attacked in 10 different trials. $\mathbf{x}_r$ refers to reference image and $\tilde{\mathbf{x}_a}$ to the corresponding reconstruction of the adversarial image $\mathbf{x}_a$. The maximum input noise perturbation levels λ are limited to $1, 3$, and 5.

Decoder quality. We further study the quality of the decoder of the proposed model. Here we evaluate the MSSSIM between the reference images x_r and its reconstructions $\tilde{x_r}$ and the adversarial images x_a and its corresponding reconstructions $\tilde{x_a}$ for both attack modes. The results are reported in Table 5.2 and 5.3. Compared to the non-robust variants (VAE, β-VAE, β-TCVAE), the quality of the reconstructions of the reference images is compromised in robust VAE models (LipschitzVAE, SE, AVAE). In contrast, the proposed model exhibits comparatively better performance. This further aligns with the observation that our model yields better reconstruction fidelity than all the baselines. Further, we observe a higher similarity between the adversarial images and their reconstructions for all robust VAE models. This is because all these models employ adversarial training.

MNIST	MSSSIM$(\mathbf{x}_r, \tilde{\mathbf{x}_r})(\uparrow)$	Latent space attack			Maximum damage attack		
		MSSSIM$(\mathbf{x}_a, \tilde{\mathbf{x}_a})(\uparrow)$			MSSSIM$(\mathbf{x}_a, \tilde{\mathbf{x}_a})(\uparrow)$		
		1	3	5	1	3	5
VAE	0.94	0.71	0.42	0.24	0.66	0.35	0.31
β-VAE	0.93	0.73	0.48	0.31	0.64	0.38	0.35
β-TCVAE	0.93	0.72	0.47	0.25	0.64	0.40	0.38
LipschitzVAE	0.85	0.70	0.46	0.38	0.66	0.44	0.39
SE	0.91	0.71	0.53	0.48	0.69	0.58	0.50
AVAE	0.92	0.70	0.55	0.50	0.71	0.62	0.59
Ours	**0.97**	**0.89**	**0.79**	**0.55**	**0.82**	**0.70**	**0.61**

FASHIONMNIST	MSSSIM$(\mathbf{x}_r, \tilde{\mathbf{x}_r})(\uparrow)$	Latent space attack			Maximum damage attack		
		MSSSIM$(\mathbf{x}_a, \tilde{\mathbf{x}_a})(\uparrow)$			MSSSIM$(\mathbf{x}_a, \tilde{\mathbf{x}_a})(\uparrow)$		
		1	3	5	1	3	5
VAE	0.88	0.61	0.52	0.50	0.45	0.33	0.29
β-VAE	0.87	0.61	0.57	0.51	0.53	0.26	0.29
β-TCVAE	0.88	0.62	0.59	0.53	0.54	0.23	0.31
LipschitzVAE	0.84	0.77	0.58	0.55	0.58	0.29	0.35
SE	0.86	0.78	0.67	0.60	0.67	0.65	0.48
AVAE	0.87	**0.80**	0.75	0.61	0.85	0.65	0.47
Ours	**0.91**	0.79	**0.76**	**0.62**	**0.89**	**0.71**	**0.56**

Table 5.2 Decoder quality - Similarity between images and their corresponding reconstructions for MNIST and FASHIONMNIST images. We consider the MSSSIM between the reference image($\mathbf{x}_r$) and its reconstruction($\tilde{\mathbf{x}_r}$) and the adversarial image($\mathbf{x}_a$) and its reconstruction($\tilde{\mathbf{x}_a}$) for both latent space and maximum damage attack. The reference image is the target image for the latent space attack, and for the maximum damage attack, the reference image is the input image.

SVHN	MSSSIM($\mathbf{x}_r, \tilde{\mathbf{x}}_r$)(↑)	Latent space attack			Maximum damage attack		
		MSSSIM($\mathbf{x}_a, \tilde{\mathbf{x}_a}$)(↑)			MSSSIM($\mathbf{x}_a, \tilde{\mathbf{x}_a}$)(↑)		
		1	3	5	1	3	5
VAE	0.85	0.71	0.62	0.57	0.68	0.58	0.50
β-VAE	0.84	0.72	0.60	0.58	0.68	0.57	0.52
β-TCVAE	0.85	0.71	0.61	0.59	0.69	0.58	0.51
LipschitzVAE	0.80	0.74	0.63	0.58	0.68	0.60	0.54
SE	0.83	0.79	0.69	0.62	0.82	0.78	0.62
Ours	**0.90**	**0.81**	**0.73**	**0.66**	**0.84**	**0.80**	**0.65**

CELEBA	MSSSIM($\mathbf{x}_r, \tilde{\mathbf{x}}_r$)(↑)	Latent space attack			Maximum damage attack		
		MSSSIM($\mathbf{x}_a, \tilde{\mathbf{x}_a}$)(↑)			MSSSIM($\mathbf{x}_a, \tilde{\mathbf{x}_a}$)(↑)		
		1	3	5	1	3	5
VAE	0.84	0.79	0.72	0.65	0.76	0.71	0.65
β-VAE	0.83	0.74	0.71	0.67	0.75	0.73	0.69
β-TCVAE	0.83	0.73	0.68	0.64	0.75	0.70	0.68
LipschitzVAE	0.79	0.74	0.70	0.67	0.75	0.72	0.69
SE	0.80	0.79	0.75	**0.70**	0.78	0.76	0.74
Ours	**0.86**	**0.80**	**0.77**	**0.70**	**0.80**	**0.79**	**0.77**

Table 5.3 Decoder quality - Similarity between images and their corresponding reconstructions for SVHN and CELEBA images. We consider the MSSSIM between the reference image($\mathbf{x}_r$) and its reconstruction($\tilde{\mathbf{x}}_r$) and the adversarial image($\mathbf{x}_a$) and its reconstruction($\tilde{\mathbf{x}}_a$) for both latent space and maximum damage attack. The reference image is the target image for the latent space attack, and for the maximum damage attack, the reference image is the input image.

Results. To compare the fidelity of the learned representations, we evaluate the FID of the generated samples as shown in Table 5.1. Overall we see that the proposed model outperforms all the considered baselines in terms of robustness and offers superior generation performance. Even for complex datasets like SVHN and CELEBA, we observe the same trend with FID and robustness measures. These results are especially promising since we did not employ any extensive hyperparameter search for training. Our results further confirm that both robust and high fidelity models are possible. Since we employ FGSM-based adversarial training, the training time required is cheaper when compared to the expensive PGD-based training used in smooth encoders (SE). The computation time for a single iteration of SEs is two times more compared to our method.

5.4.4 Ablation Study

In this section, we compare three different variants of regularized deterministic autoencoders to evaluate the importance of joint regularization of the original and adversarial samples. We begin with the model proposed in Chapter 4, which we denote as GMM-DAE. Second, we study the augmented model defined by equations (5.2) and (5.3) in Section 5.3.2, but without the coupling of original and adversarial latent representations (Augmented). We compare the robustness and fidelity of these models with our proposed model, i.e., where original and adversarial latent representations are not only regularized towards the same prior but coupled according to equations (5.4), (5.5) and (5.6) in Section 5.3.3 (Ours). The observed metrics are reported in Table 5.4. We observe that the proposed method yields comparatively better performance in terms of robustness while still maintaining the generation fidelity when compared to the non-robust version, GMM-DAE. It can also be seen that enforcing coupling between the latent representations of the original and adversarial samples (Ours) leads to better performance than simply augmenting them (Augmented). It is worth pointing out that the GMM-DAE maintains better performance than a standard VAE model (Table 5.1, MNIST VAE results). This is due to the well-structured latent space in GMM-DAE. This observation further confirms our hypothesis in Section 5.3.1 that robustness can be improved when similar-looking samples are modeled together in the model's latent space.

Method	Latent space attack			Maximum damage attack			FID(↓)
	MSSSIM($x_r, \tilde{x_a}$)(↓)			MSSSIM($\tilde{x_r}, \tilde{x_a}$)(↑)			
	1	3	5	1	3	5	
GMM-DAE [161]	0.54	0.70	0.82	0.75	0.37	0.30	**38.89**
Augmented	0.47	0.59	0.71	0.79	0.56	0.54	40.16
Ours	**0.38**	**0.47**	**0.60**	**0.92**	**0.82**	**0.69**	39.37

Table 5.4 Ablation study on MNIST images. Augmented refers to the model definition in eqs (5.2) and (5.3). Here x_r refers to reference image, x_a to adversarial image and $\tilde{x_r}$, $\tilde{x_a}$ to their corresponding reconstructions. The maximum input noise perturbation level λ is limited to $1, 3$ and 5.

5.4.5 Hyperparameter Sensitivity Analysis

In this section, we conduct a sensitivity study of the model robustness to the number of components in the chosen GMM prior and that of the coupling strength parameter α in MNIST images.

Number of modes	Latent space attack						Maximum damage attack						FID(↓)
	MSSSIM($\mathbf{x}_r, \tilde{\mathbf{x}}_a$)(↓)			MSSSIM($\mathbf{x}_r, \mathbf{x}_a$)(↓)			MSSSIM($\tilde{\mathbf{x}}_r, \tilde{\mathbf{x}}_a$)(↑)			MSSSIM($\mathbf{x}_r, \mathbf{x}_a$)(↓)			
	1	3	5	1	3	5	1	3	5	1	3	5	
1	0.48	0.62	0.70	0.99	0.93	0.86	0.89	0.73	0.62	0.98	0.92	0.86	42.45
5	0.43	0.53	0.66	0.98	0.92	0.83	0.91	0.76	0.65	0.98	0.91	0.82	40.11
10	0.38	0.47	0.60	0.95	0.90	0.80	0.92	0.82	0.69	0.98	0.89	0.78	39.37
15	0.37	0.45	0.58	0.95	0.89	0.80	0.93	0.82	0.70	**0.97**	0.89	0.77	39.04
20	0.37	0.45	**0.57**	**0.94**	**0.88**	0.79	0.93	**0.83**	**0.71**	**0.97**	**0.88**	0.78	38.49
25	**0.36**	**0.43**	0.58	**0.94**	**0.88**	**0.78**	**0.94**	**0.83**	**0.71**	**0.97**	**0.88**	**0.77**	**38.02**

Table 5.5 Sensitivity analysis of the number of modes in the GMM prior on MNIST images.

Number of modes in the GMM prior. The number of modes in the chosen prior is a hyperparameter of the proposed model. Hence we report a sensitivity analysis of the number of modes of the GMM prior and the observed robustness of the model. We analyze our model's robustness and generation performance on MNIST images for different components in the chosen prior in Table 5.5. We use the same number of modes used in the previous chapter for all our experiments. As expected from previous analysis, with an increased number of modes in the GMM prior, the generation performance of our extended model also improved. Most importantly, we observe a similar trend for robustness as well. The model exhibits improved robustness with more components in the chosen GMM prior.

Coupling strength. In this section, we study the sensitivity of our model towards the coupling strength α (see Table 5.6). We observe that a more significant coupling strength α leads to improved robustness against latent space and maximum damage attacks. However, as mentioned in the limitations section, a strong coupling strength, i.e., $\alpha = 1$, compromises the generation fidelity of the model. In our experiments, we tuned the coupling strength on each dataset. We observed that a coupling strength in the range of $0.9 \leq \alpha < 1$ yields the best trade-off between generation and robustness across all datasets. In our experiments, we chose $\alpha = 0.95$ for MNIST and FASHIONMNIST images, and $\alpha = 0.92$ for SVHN and CELEBA images.

5.4.6 Robustness to Downstream Applications

Since the learned representations of VAEs are often used for various downstream tasks, it is also vital to verify how adversarial attacks affect the performance of the same. To showcase the effect of adversarial attacks on downstream classification tasks, we train an MLP classifier on the latent space of the model and observe the accuracy drop in the presence of the latent

Coupling parameter	Latent space attack						Maximum damage attack						FID(↓)
α	MSSSIM$(\mathbf{x}_r, \tilde{\mathbf{x}}_a)(\downarrow)$			MSSSIM$(\mathbf{x}_r, \mathbf{x}_a)(\downarrow)$			MSSSIM$(\tilde{\mathbf{x}}_r, \tilde{\mathbf{x}}_a)(\uparrow)$			MSSSIM$(\mathbf{x}_r, \mathbf{x}_a)(\downarrow)$			
	1	3	5	1	3	5	1	3	5	1	3	5	
0.1	0.46	0.56	0.68	0.98	0.91	0.84	0.80	0.59	0.50	0.98	0.91	0.80	40.18
0.3	0.44	0.55	0.65	0.97	0.92	0.82	0.81	0.65	0.61	0.97	0.90	0.80	39.84
0.5	0.43	0.53	0.64	0.97	0.90	0.82	0.85	0.75	0.63	0.98	0.89	0.80	40.01
0.7	0.40	0.49	0.63	0.96	0.90	0.81	0.90	0.79	0.67	0.98	0.89	0.79	39.28
0.9	0.39	0.48	0.62	0.95	0.90	0.80	0.91	0.81	0.68	0.98	0.89	0.78	39.61
0.95	**0.38**	**0.47**	0.60	**0.95**	0.90	**0.80**	**0.92**	0.82	**0.69**	0.98	**0.89**	**0.78**	**39.37**
1.0	**0.38**	**0.47**	**0.59**	**0.95**	**0.89**	**0.80**	**0.92**	**0.83**	**0.69**	**0.97**	**0.87**	**0.78**	41.86

Table 5.6 Sensitivity analysis of the hyperparameter alpha on MNIST images.

Method	MNIST		FASHIONMNIST		SVHN	
	clean acc.(↑)	$\lambda = 1$(↑)	clean acc.(↑)	$\lambda = 1$(↑)	clean acc.(↑)	$\lambda = 1$(↑)
VAE	92.16	58.85	80.65	59.15	61.70	25.63
β-TCVAE	93.02	61.06	81.25	60.77	62.02	30.12
LipschitzVAE	90.78	62.00	80.50	62.06	60.99	33.99
SE	93.81	68.65	80.10	66.83	62.36	44.40
Ours	**96.08**	**91.78**	**85.96**	**78.86**	**70.96**	**59.20**

Table 5.7 Robustness of downstream classifier trained in the latent space of the model under adversarial attack - we report the clean accuracy and the accuracy during attack defined in eqn(5.11), for $\lambda = 1$.

space attacks for a constrained noise norm $\lambda = 1$. The observed values and clean accuracy are shown in Table 5.7. The classifier trained on the latent space of the proposed model achieves better accuracy when compared to the baseline models under latent space attack.

5.4.7 Network Architecture and Implementation Details

Network architectures. We use a consistent network architecture for the encoder-decoder pair during training. For MNIST and FASHIONMNIST images, we train a fully connected network architecture, a 4-layer multi-layer perceptron (MLP) with 200 neurons and ReLU activation at each layer. For SVHN and CELEB images, we use a convolution network architecture similar to the last chapter mentioned in Section 4.4.6. The encoder includes a 4-layer convolution network with the number of output channels $(128, 256, 512, 1024)$, respectively, with strides equal to 2 and a kernel size of $(4, 4)$. And the decoder comprises a 4-layer de-convolution network with the number of output channels $(1024, 512, 256, 128)$

respectively, with strides equal to 2 and a kernel size of $(4,4)$. The latent space dimension of 10 is used for MNIST and FASHIONMNIST images, 100 for SVHN, and 64 for CELEBA images.

Implementation details. We carried out all the experiments on a single GTX1080 GPU with 16 GB RAM. All the conducted experiments were part of a carbon-neutral framework-based GPU cluster and did not contribute to climate change.

For training the encoder-decoder network of our model, we utilize an ADAM [105] optimizer with a batch size of 100 and an initial learning rate of 0.002 with exponential decay based on the validation loss. We follow the same setup as in the last chapter for the multi-modal prior definition. The coupling parameter is 0.95 for MNIST and FASHIONMNIST images and 0.92 for SVHN and CELEBA images. For the classification downstream application, we train a simple two-layer MLP-based classifier. The network is trained for 25 epochs with an ADAM [105] based optimizer with a learning rate of 0.01, batch size of 100.

We use similar architecture for all the baseline models considered for a fair comparison. We used the Pytorch implementation in the Githib repository [] for training the baseline models, VAE [108], $\beta-$VAE [79] and $\beta-$TCVAE [26] . For LipschitzVAE, we used the official Pytorch implementation [11]. For SE [23], we re-implemented the method in Pytorch, and for AVAE [22], we reimplemented a Pytorch version of the official JAX-based version. Since the official GitHub implementation for AVAEs only provides an MLP-based training pipeline, we only report AVAE results for MNIST and FASHIONMNIST images.

Evaluation setup. To evaluate the model's robustness, we mainly consider two types of adversarial attacks, latent space attacks and maximum damage attacks. Under these attacks, we consider 100 randomly chosen test images from the corresponding dataset for experimental analysis and run 10 simulations to report the results. The noise perturbation levels of $1,3$ and 5 are chosen. While choosing the target image for latent space attacks, we explicitly choose an image from a different class than of original image for MNIST, FASHIONMNIST, and SVHN images. To evaluate the learned representations' fidelity, we report Fréchet Inception Distance (FID) [78] of the generated samples. We calculate the FID between 10000 generated images and validation images for the corresponding dataset and report the average value obtained after five different runs. The FIDs observed for the proposed method and error bars (for different runs) are as follows, MNIST: 39.37 ± 0.9, FASHION-MNIST: 64.89 ± 0.9, SVHN: 38.89 ± 1.2 and CELEBA: 51.98 ± 1.3.

5.5 Conclusion

Developing robust VAE models is crucial since the learned representations of VAEs are frequently used for various applications. Motivated by the recent research towards deterministic alternatives to VAEs, we study the robustness of deterministic autoencoders in this chapter. We extend recently developed regularization schemes to efficiently couple the adversarial examples and the learned representations during training. Our experiments show that adversarially trained multimodal deterministic autoencoders offer significantly improved adversarial robustness and high fidelity in the learned latent space with proper regularization.

Chapter 6

Conclusion

In this chapter, we review the main findings and contributions of the methods presented in this thesis. We then discuss some of the limitations associated with the proposed approaches and possible future work.

6.1 Discussion

In this thesis, we investigated the learned representations of two popular deep generative models, Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), and proposed methods to optimize the latent space of these models.

We addressed the controllability of the learned latent space of GANs in Chapter 3. Regulating the content of the generated images merely based on the numerosity of the objects is a complex problem. We found that the proposed architectural modifications to the state-of-the-art StyleGAN2 network generated images of very high quality even under challenging scenarios, i.e., no specification of the spatial layout and a limited amount of training data. Our experiments also show that the number of objects in the images provides vital information regarding their distinguishability during feature learning, allowing us to control the image generation process. The existing real-world open-source datasets are not directly usable or derivable for our task, as the number of images with a given number of objects from consistent classes is minimal. To overcome this, we derived the Citycount dataset for our evaluation, and we believe that such a dataset will be an asset for further research in the community.

Variational autoencoders enable learning meaningful representations of complex high-dimensional data without any supervision. This further enhances the usability of these learned representations for various downstream tasks and applications where limited data is available, e.g., due to privacy/security concerns. Therefore, it is critical to investigate

the robustness of these representations along with their accuracy, especially when used in real-world applications. In Chapter 4, we first explored some practical limitations associated with VAEs and discussed deterministic alternatives to the variational formulation. Motivated by recent developments towards deterministic autoencoders[60] from stochastic VAEs for data generation, we proposed a novel regularization scheme to enable simple and end-to-end training of regularized deterministic autoencoders. The recent discovery of the vulnerability of learned representations in VAEs is concerning, as this calls into question the generalization of learned latent space in VAEs. In reviewing related work in this area, we found that there is a tradeoff between robustness and fidelity. Therefore, in Chapter 5, we took a step towards more robust models and proposed a method to train robust regularized deterministic autoencoders with high fidelity. The proposed approach can be readily applied to effectively structure the latent space of existing autoencoding frameworks towards multi-modal Gaussian priors. Currently, there is limited work in this direction, and our method encourages potential future work to develop robust VAE models.

6.2 Limitations and Future Work

In Chapter 3, we saw that the proposed GAN places the objects in the images in a reasonable spatial arrangement without providing additional information, such as the bounding box information of the objects. Although we have demonstrated the potential of the model in a challenging scenario such as the CityCount dataset, we have yet to consider a highly complex environment where several objects are highly occluded and where certain objects are only partially visible in the images. Although we expect the model to perform well in such a scenario, a major limitation would be the availability of a dataset with count annotations. A semi-supervised training approach that incorporates bounding box and count information for specific images may be required to achieve comparable performance in such a setting. In the empirical study, we also examined the model's potential to extrapolate between classes. The ability to generalize beyond the range of the count value of an object class is highly interesting and challenging future work in this area [184].

One limitation of the proposed regularized deterministic autoencoders in Chapter 4 is the necessity to choose the prior distribution in advance. We showed that fixing a suitable number of modes for the GMM is important to provide better sampling quality. Also, by considering marginal CDFs, we simplified the original distance metric from the KS test. While reducing computational complexity during training, this comes at the cost of an additional loss term. Further, our proposed addition of the KS distance is not suitable for matching higher-order moments of the latent representations to the target prior, which, at least from a conceptual

point of view, can lead to a mismatch to the prior. Further, our loss only facilitates matching empirical marginal CDFs of latent representations to the marginal CDFs of the prior evaluated latent vectors. Consequently, our regularization loss might be a less stable training signal for small batch sizes in high dimensions.

The choice of the MSE for covariance matching loss is purely based on its prevalence in the literature. Additionally, all three loss terms (reconstruction loss, KS distance loss, and the covariance matching loss) behave similarly, as they are all squares. While we did not investigate any other metrics for matrix comparison in this scenario, exploring other options for covariance matching is an interesting area for future studies [58, 54]. Further, we have not considered the case where there exists a class imbalance in the dataset. We would expect the model to separate the classes if the imbalance is weak and the classes are sufficiently different such that the reconstruction loss outweighs the regularization penalty for the mismatch. Extending our prior to accommodate this by introducing a weighted GMM prior is also an interesting direction for future work. Another potential solution would be to design a training scheme to learn the weights of the GMM components during training. Since the model performs well in unsupervised clustering experiments, the proposed regularizer could be potentially applied to structure the feature space of the classifier to enhance the classification performance. Further, it would be worth investigating the potential of the multi-modal regularization scheme to optimize the learned feature space of recently proposed models such as Latent diffusion models [157].

The proposed adversarial training scheme to enhance the robustness of the deterministic autoencoder in Chapter 5; although cheaper compared to the current adversarially trained robust models, the FGSM-based training scheme is still expensive when compared to the non-robust counterpart. The coupling parameter α introduced in the proposed method is an additional hyperparameter to tune, and enforcing strong coupling, i.e., $\alpha = 1$, might compromise the generation fidelity of the resulting model. While we need the coupling term to perform adversarial training, the overly strong coupling will necessarily lead to deteriorated reconstructions, similar to VAEs. It is worth noting that we did not perform intensive hyperparameter optimization in this line of experiments. Hyperparameter optimization might be needed while adapting the method to other datasets. Further, it would be interesting to explore the impact of the adversarial attacks in other potential downstream applications, such as high-dimensional black-box optimization in the latent space of VAE models.

We should also consider this research's possible negative social impact, especially in safety-critical applications. Although we observe superior robustness in our model against the existing attacks, similar performance cannot be guaranteed on newly discovered attacks on VAEs. Hence, when deployed in real-world applications, we highly recommend testing

the model continuously against newly designed attacks. We also urge the machine learning community to responsibly pursue this work to enable potential future research without misuse.

Bibliography

[1] R. Abdal, P. Zhu, N. J. Mitra, and P. Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. ACM Transactions on Graphics (TOG), 40:1 – 21, 2020. 21

[2] V. Agarwal, R. Shetty, and M. Fritz. Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9687–9695, 2020. 30

[3] D. Alexandre, C. Chang, W. Peng, and H. Hang. An autoencoder-based learned image compressor: Description of challenge proposal by nctu. ArXiv, abs/1902.07385, 2019. 16

[4] L. Amsaleg, J. Bailey, A. Barbe, S. M. Erfani, T. Furon, M. E. Houle, M. Radovanović, and X. V. Nguyen. High intrinsic dimensionality facilitates adversarial attack: Theoretical evidence. IEEE Transactions on Information Forensics and Security, 16:854–865, 2021. 81

[5] N. Anand and P. Huang. Generative modeling for protein structures. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems 31, pages 7494–7505. Curran Associates, Inc., 2018. 28

[6] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In 2015 IEEE International Conference on Computer Vision (ICCV), pages 2425–2433, 2015. 30

[7] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In International Conference on Machine Learning, 2017. 14

[8] Y. Balaji, M. Min, B. Bai, R. Chellappa, and H. Graf. Conditional gan with discriminative filter generation for text-to-video synthesis. pages 1995–2001, 08 2019. 29

[9] P. Baldi. Autoencoders, unsupervised learning and deep architectures. In Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27, UTLW'11, page 37–50. JMLR.org, 2011. 16

[10] L. Baringhaus and N. Henze. A consistent test for multivariate normality based on the empirical characteristic function. Metrika, 35:339–348, 1988. 54

[11] B. Barrett, A. Camuto, M. Willetts, and T. Rainforth. Certifiably robust variational autoencoders. In NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications, 2021. 78, 79, 85, 101

[12] G. Batzolis, J. Stanczuk, C. Schönlieb, and C. Etmann. Conditional image generation with score-based diffusion models. ArXiv, abs/2111.13606, 2021. 22

[13] Y. Bengio, A. C. Courville, and P. Vincent. Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 35(8):1798–1828, August 2013. 2

[14] V. Berger and M. Sebag. Variational Auto-Encoder: not all failures are equal. preprint, February 2020. 18, 19, 52

[15] A. K. Bhunia, S. Khan, H. Cholakkal, R. M. Anwer, F. S. Khan, and M. Shah. Handwriting transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 1086–1094, October 2021. 21

[16] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. Journal of the American Statistical Association, 112:859 – 877, 2016. 17

[17] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(11):7327–7347, 2022. 18, 25

[18] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. In International Conference on Learning Representations (ICLR), 2018. 3, 15, 16

[19] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, and A. Askell et al. Language models are few-shot learners. In Advances in Neural Information Processing Systems, volume 33, pages 1877–1901, 2020. 1, 21

[20] A. Camuto, M. Willetts, S. J. Roberts, C. C. Holmes, and T. Rainforth. Towards a theoretical understanding of the robustness of variational autoencoders. In AISTATS, 2021. 79

[21] F. P. Casale, A. V Dalca, L. Saglietti, J. Listgarten, and N. Fusi. Gaussian process prior variational autoencoders. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, page 10390–10401, Red Hook, NY, USA, 2018. Curran Associates Inc. 19

[22] T. A. Cemgil, S. Ghaisas, K. Dvijotham, S. Gowal, and P. Kohli. The autoencoding variational autoencoder. In Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, 2020. 1, 2, 78, 79, 85, 101

[23] T. A. Cemgil, S. Ghaisas, K. (Dj) Dvijotham, and P. Kohli. Adversarially robust representations with smooth encoders. In International Conference on Learning Representations, 2020. 1, 2, 78, 79, 85, 101

[24] A. B. Chan, Z. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In 2008 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–7, 2008. 30

[25] M. Chen, A. Radford, J. Wu, H. Jun, P. Dhariwal, D. Luan, and I. Sutskever. Generative pretraining from pixels. In International Conference on Machine Learning, 2020. 21

[26] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud. Isolating sources of disentanglement in vaes. In NIPS, page 2615–2625, 2018. 18, 85, 101

[27] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In NIPS, 2016. 29, 46

[28] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational lossy autoencoder. ArXiv, abs/1611.02731, 2017. 19

[29] X. Chen, N. Mishra, M. Rohaninejad, and P. Abbeel. Pixelsnail: An improved autoregressive generative model. ArXiv, abs/1712.09763, 2017. 11

[30] Y. Chen, F. Shi, A. G. Christodoulou, Z. Zhou, Y. Xie, and D. Li. Efficient and accurate mri super-resolution using a generative adversarial network and 3d multi-level densely connected network. In MICCAI, 2018. 4, 33

[31] Z. Chen, C.K. Yeo, B. S. Lee, and C. T. Lau. Autoencoder-based network anomaly detection. In 2018 Wireless Telecommunications Symposium (WTS), pages 1–5, 2018. 16

[32] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 28, 37

[33] F. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah. Diffusion models in vision: A survey. ArXiv, abs/2209.04747, 2022. 22

[34] B. Dai and D. Wipf. Diagnosing and enhancing VAE models. In International Conference on Learning Representations, 2019. 62, 78

[35] S. Dehaene. The number sense: How the mind creates mathematics. Oxford University Press, 2011. 27, 28

[36] L. Deng. The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine, 29(6):141–142, 2012. 23, 28, 35, 36, 62

[37] J. Devlin, M. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. ArXiv, abs/1810.04805, 2019. 21

[38] P. Dhariwal and A. Q. Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, Advances in Neural Information Processing Systems, 2021. 22

[39] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. CoRR, abs/1611.02648, 2016. 18, 53, 62

[40] Z. Ding, Y. Xu, W. Xu, G. Parmar, Y. Yang, M. Welling, and Z. Tu. Guided variational autoencoder for disentanglement learning. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 7917–7926, 2020. 51

[41] L. Dinh, D. Krueger, and Y. Bengio. Nice: Non-linear independent components estimation. CoRR, abs/1410.8516, 2014. 21

[42] L. Dinh, J. N. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. ArXiv, abs/1605.08803, 2016. 21

[43] K. G. Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In Proceedings of the IEEE international conference on computer vision, pages 5736–5745, 2017. 53

[44] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9185–9193, jun 2018. 80

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations, 2021. 21

[46] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 29. Curran Associates, Inc., 2016. 18

[47] Y. Du and I. Mordatch. Implicit generation and generalization in energy-based models. ArXiv, abs/1903.08689, 2019. 11

[48] E. Dupont. Learning disentangled joint continuous and discrete representations. In NeurIPS, 2018. 66

[49] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 12868–12878, 2020. 21, 22

[50] P. Esser, E. Sutter, and B. Ommer. A variational u-net for conditional appearance and shape generation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8857–8866, 2018. 29

[51] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer. Multiscale vision transformers. In ICCV, 2021. 21

[52] G. Fasano and A. Franceschini. A multidimensional version of the Kolmogorov-Smirnov test. Monthly Notices of the Royal Astronomical Society, 225:155–170, 1987. 54

[53] L. Fiaschi, U. Koethe, R. Nair, and F. A. Hamprecht. Learning to count with regression forest and structured labels. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), pages 2685–2688, 2012. 30

[54] W. Förstner and B. Moonen. A Metric for Covariance Matrices, pages 299–309. Springer Berlin Heidelberg, 2003. 105

[55] Q. Fournier and D. Aloise. Empirical comparison between autoencoders and traditional dimensionality reduction methods. In 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), pages 211–214, 2019. 16

[56] N. Gählert, N. Jourdan, M. Cordts, U. Franke, and J. Denzler. Cityscapes 3d: Dataset and benchmark for 9 dof vehicle detection. CoRR, abs/2006.07864, 2020. 37

[57] R. Gandikota and N. Brown. Pro-ddpm: Progressive growing of variable denoising diffusion probabilistic models for faster convergence. In 33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022, page 121. BMVA Press, 2022. 22

[58] C. Garcia. A simple procedure for the comparison of covariance matrices. BMC evolutionary biology, 12:222, 11 2012. 105

[59] P. Ghosh, A. Losalka, and M. J. Black. Resisting adversarial attacks using gaussian mixture variational autoencoders. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. 77, 80, 81

[60] P. Ghosh, M. S. M. Sajjadi, A. Vergari, M. Black, and B. Scholkopf. From variational to deterministic autoencoders. In International Conference on Learning Representations, 2020. xvii, 1, 2, 5, 18, 19, 52, 55, 62, 69, 73, 78, 104

[61] G. Giannone, D. Nielsen, and O. Winther. Few-shot diffusion models. ArXiv, abs/2205.15463, 2022. 22

[62] M. V. Giuffrida, H. Scharr, and S. A. Tsaftaris. Arigan: Synthetic arabidopsis plants using generative adversarial network. In 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pages 2064–2071, 2017. 30

[63] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. Journal of the American Statistical Association, 102:359 – 378, 2007. 54

[64] L. Gondara. Medical image denoising using convolutional denoising autoencoders. In 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), pages 241–246, 2016. 16

[65] G. Gondim-Ribeiro, P. Tabacof, and E. Valle. Adversarial attacks on variational autoencoders. ArXiv, abs/1806.04646, 2018. 78, 79, 85

[66] M. Gong, Y. Xu, C. Li, K. Zhang, and K. Batmanghelich. Twin auxiliary classifiers gan. Advances in neural information processing systems, 32:1328–1337, 12 2019. 29, 46

[67] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press. 2, 11, 12, 13

[68] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. 80

[69] E. Gosset. A three-dimensional extended Kolmogorov-Smirnov test as a useful tool in astronomy. Astronomy and Astrophysics., 188(1):258–264, 1987. 54

[70] P. Goyal, Z. Hu, X. Liang, C. Wang, E. P. Xing, and C. Mellon. Nonparametric variational auto-encoders for hierarchical representation learning. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 5104–5112, 2017. 19

[71] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In Advances in neural information processing systems (NeurIPS), pages 5767–5777, 2017. 14, 15

[72] X. Guo, X. Liu, E. Zhu, and J. Yin. Deep clustering with convolutional autoencoders. In Neural Information Processing, 2017. 53

[73] R. Gómez-Bombarelli, D. Duvenaud, J. Hernández-Lobato, J. Aguilera-Iparraguirre, T. Hirzel, R. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. ACS Central Science, 4, 2016. 53, 69, 70, 74

[74] D. R. Ha and J. Schmidhuber. World models. ArXiv, abs/1803.10122, 2018. 77

[75] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao. A survey on vision transformer. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(1):87–110, 2023. 21

[76] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen. Attgan: Facial attribute editing by only changing what you want. IEEE Transactions on Image Processing, 28(11):5464–5478, Nov 2019. 29

[77] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 31–35, March 2016. 53

[78] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In NIPS, 2017. 25, 62, 74, 85, 101

[79] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In ICLR, 2017. 18, 85, 101

[80] I. Higgins, A. Pal, A. Rusu, L. Matthey, C. Burgess, A. Pritzel, M. Botvinick, C. Blundell, and A. Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. In Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, page 1480–1490. JMLR.org, 2017. 77

[81] T. Hinz, S. Heinrich, and S. Wermter. Generating multiple objects at spatially distinct locations. In International Conference on Learning Representations, 2019. 29

[82] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. ArXiv, abs/1902.00275, 2019. 1, 11, 21

[83] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. ArXiv, abs/2006.11239, 2020. 11, 22

[84] J. Ho, T. Salimans, A. A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video diffusion models. In ICLR Workshop on Deep Generative Models for Highly Structured Data, 2022. 22

[85] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2261–2269, 2017. 32, 33

[86] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017. 29

[87] X. Huang and S. J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. 2017 IEEE International Conference on Computer Vision (ICCV), pages 1510–1519, 2017. 31

[88] J. Irwin, T. Sterling, Michael M. Mysinger, Erin S. Bolstad, and R. Coleman. Zinc: A free tool to discover chemistry for biology. Journal of Chemical Information and Modeling, 52:1757 – 1768, 2012. 68

[89] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5967–5976, 2016. 29

[90] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine. Planning with diffusion for flexible behavior synthesis. In International Conference on Machine Learning, 2022. 1

[91] Y. Jeong and H. O. Song. Learning discrete and continuous factors of data via alternating disentanglement. In International Conference on Machine Learning (ICML), 2019. 66

[92] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid. Deep subspace clustering networks. In Advances in Neural Information Processing Systems, pages 24–33, 2017. 53

[93] Y. Jiang, S. Chang, and Z. Wang. Transgan: Two transformers can make one strong gan. ArXiv, abs/2102.07074, 2021. 21, 22

[94] L. Jin, F. Doshi-Velez, T. Miller, L. Schwartz, and W. Schuler. Unsupervised learning of pcfgs with normalizing flow. In Annual Meeting of the Association for Computational Linguistics, 2019. 21

[95] W. Jin, R. Barzilay, and T. Jaakkola. Junction tree variational autoencoder for molecular graph generation. In ICML, 2018. 51

[96] J. Johnson, B. Hariharan, L. Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. B. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1988–1997, 2017. 28, 35, 36

[97] M. Kang and J. Park. Contragan: Contrastive learning for conditional image generation. arXiv: Computer Vision and Pattern Recognition, 2020. 29, 41

[98] A. Karnewar and O. Wang. Msg-gan: Multi-scale gradients for generative adversarial networks. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 7796–7805, 2020. 32

[99] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In International Conference on Learning Representations, 2018. 1, 3, 15, 16, 29, 30

[100] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila. Training generative adversarial networks with limited data. In Proc. NeurIPS, 2020. xvii, 1, 30, 32, 37, 42

[101] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 3, 15, 16, 29, 30

[102] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 8107–8116, 2020. xii, 1, 3, 15, 28, 30, 31, 37, 41

[103] S. H. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah. Transformers in vision: A survey. ACM Computing Surveys (CSUR), 54:1 – 41, 2021. 22

[104] I. Kim, S. J. Han, J. Baek, S. Park, J. Han, and J. Shin. Quality-agnostic image recognition via invertible decoder. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 12252–12261, 2021. 21

[105] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. 37, 73, 101

[106] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In Advances in Neural Information Processing Systems, volume 31. Curran Associates, Inc., 2018. 21

[107] D. P. Kingma and M. Welling. Auto-encoding variational bayes. CoRR, abs/1312.6114, 2014. 2, 11, 18, 51, 62

[108] D. P. Kingma and M. Welling. An introduction to variational autoencoders. Found. Trends Mach. Learn., 12:307–392, 2019. 85, 101

[109] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In International Conference on Learning Representations, 2021. 1, 22

[110] J. Kos, I. Fischer, and D. Song. Adversarial examples for generative models. 2018 IEEE Security and Privacy Workshops (SPW), pages 36–42, 2018. 78, 79, 85

[111] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher. Ask me anything: Dynamic memory networks for natural language processing. In Proceedings of The 33rd International Conference on Machine Learning, volume 48 of Proceedings of Machine Learning Research, pages 1378–1387, New York, New York, USA, 2016. PMLR. 30

[112] M. Kumar, M. Babaeizadeh, D. Erhan, C. Finn, S. Levine, L. Dinh, and D. Kingma. Videoflow: A conditional flow-based model for stochastic video generation. In International Conference on Learning Representations, 2020. 21

[113] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. ArXiv, abs/1607.02533, 2016. 80

[114] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato. Grammar variational autoencoder. In Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, page 1945–1954, 2017. xvii, 53, 68, 70, 74

[115] A. Kuzina, M. Welling, and M. J. Tomczak. Diagnosing vulnerability of variational auto-encoders to adversarial attacks. RobustML Workshop@ICLR 2021, 2021. 78, 91

[116] D. B. Lee, D. Min, S. Lee, and S. J. Hwang. Meta-GMVAE: Mixture of gaussian VAE for unsupervised meta-learning. In International Conference on Learning Representations, 2021. 19

[117] G. Li, Y. Wei, Y. Chi, Y. Gu, and Y. Chen. Breaking the sample size barrier in model-based reinforcement learning with a generative model. Advances in neural information processing systems, 33:12861–12872, 2020. 1

[118] X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. Hashimoto. Diffusion-LM improves controllable text generation. In Advances in Neural Information Processing Systems, 2022. 22

[119] M. Liu, Y. Ding, M. Xia, X. Liu, E. Ding, W. Zuo, and S. Wen. Stgan: A unified selective transfer network for arbitrary image attribute editing. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 29

[120] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In Proceedings of International Conference on Computer Vision (ICCV), December 2015. 24, 62

[121] R. Lopes, I. Reid, and P. Hobson. The two-dimensional kolmogorov-smirnov test. XI International Workshop on Advanced Computing and Analysis Techniques in Physics Research, 2007. 54

[122] X. Lu, J. Gonzalez, Z. Dai, and N. Lawrence. Structured variationally auto-encoded optimization. In Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 3267–3275, 2018. 53

[123] S. Luo, C. Shi, M. Xu, and J. Tang. Predicting molecular conformation via dynamic graph score matching. In Advances in Neural Information Processing Systems, volume 34, pages 19784–19795, 2021. 1, 22

[124] L. Maaløe, M. Fraccaro, V. Liévin, and O. Winther. Biva: A very deep hierarchy of latent variables for generative modeling. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019. 18, 19

[125] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In International Conference on Learning Representations, 2018. 78, 79, 81

[126] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In International Conference on Learning Representations, 2018. 80

[127] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial autoencoders. ArXiv, abs/1511.05644, 2015. 19

[128] K. V. Mardia. Measures of multivariate skewness and kurtosis with applications. Biometrika, 57(3):519–530, 12 1970. 54

[129] B. Mazoure, T. V. Doan, A. D., R. D. Hjelm, and J. Pineau. Leveraging exploration in off-policy algorithms via normalizing flows. In Conference on Robot Learning, 2019. 21

[130] M.Caron, B. Piotr, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In European Conference on Computer Vision, 2018. 53

[131] N. Miao, E. Mathieu, N. Siddharth, Y. W. Teh, and T. Rainforth. Intel-vaes: Adding inductive biases to variational auto-encoders via intermediary latents. ArXiv, abs/2106.13746, 2021. 19

[132] Y. Miao, L. Yu, and P. Blunsom. Neural variational inference for text processing. In Proceedings of The 33rd International Conference on Machine Learning, Proceedings of Machine Learning Research, pages 1727–1736, 2016. 51

[133] M. Mirza and S. Osindero. Conditional generative adversarial nets. ArXiv, abs/1411.1784, 2014. 15, 29, 46

[134] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957, 2018. 15, 20, 29, 41

[135] T. Miyato and M. Koyama. cgans with projection discriminator. ArXiv, abs/1802.05637, 2018. 29

[136] M. Mosbach, M. Andriushchenko, T. Alexander Trost, Matthias Hein, and Dietrich Klakow. Logit pairing methods can fool gradient-based attacks. ArXiv, abs/1810.12042, 2018. 80

[137] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. NIPS, 2011. 24, 28, 35, 36, 62

[138] A. Q. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA, volume 162 of Proceedings of Machine Learning Research, pages 16784–16804. PMLR, 2022. 1, 22

[139] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In ICML, 2017. 29, 46

[140] S. Oh, Y. Jung, S. Kim, I. Lee, and N. Kang. Deep Generative Design: Integration of Topology Optimization and Generative Models. Journal of Mechanical Design, 141(11), 09 2019. 111405. 2, 28

[141] A. Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. In NIPS, 2017. 18, 19

[142] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. M. Shazeer, A. Ku, and D. Tran. Image transformer. In International Conference on Machine Learning, 2018. 21

[143] J. A. Peacock. Two-dimensional goodness-of-fit testing in astronomy. Monthly Notices of the Royal Astronomical Society, 202(3):615–627, 1983. 54

[144] V. Prasad, D. Das, and B. Bhowmick. Variational clustering: Leveraging variational autoencoders for image clustering. CoRR, abs/2005.04613, 2020. 18, 53

[145] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin. Variational autoencoder for deep learning of images, labels and captions. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 29. Curran Associates, Inc., 2016. 51

[146] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR, abs/1511.06434, 2015. 15

[147] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. OpenAI blog, 2018. 1, 21

[148] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019. 1, 21

[149] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. ArXiv, abs/2204.06125, 2022. 1, 22

[150] A. Ramesh, M. Pavlov, G. Goh, and S. Gray. Dall·e: Creating images from text. OpenAI Blog, 2021. 22

[151] A. Razavi, A. Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. ArXiv, abs/1906.00446, 2019. 1, 18, 19

[152] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In International Conference on Machine Learning (ICML), 2016. 29

[153] M. Ren and R. S. Zemel. End-to-end instance segmentation with recurrent attention. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 293–301, 2017. 30

[154] D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. In International Conference on Machine Learning, 2015. 11, 21

[155] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML'14, page II–1278–II–1286. JMLR.org, 2014. 18, 19

[156] A. Rives, J. Meier, T. Sercu, Z. Lin, J. Liu, D. Guo, M. Ott, C. Zitnick, J. Ma, and R. Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proceedings of the National Academy of Sciences, 118:e2016239118, 04 2021. 21

[157] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 10674–10685, 2021. 1, 22, 105

[158] B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. ArXiv, abs/1511.08250, 2016. 30

[159] L. Ruan, Y. Ma, H. Yang, H. He, B. Liu, J. Fu, N. J. Yuan, Q. Jin, and B. Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. arXiv preprint, December 2022. 1

[160] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 29. Curran Associates, Inc., 2016. 25

[161] A. Saseendran, K. Skubch, S. Falkner, and M. Keuper. Shape your space: A gaussian mixture regularization approach to deterministic autoencoders. In Advances in Neural Information Processing Systems, 2021, volume 34, pages 7319–7332, 2021. xi, 2, 3, 6, 8, 51, 98

[162] A. Saseendran, K. Skubch, S. Falkner, and M. Keuper. Trading off image quality for robustness is not necessary with regularized deterministic autoencoders. In Advances in Neural Information Processing Systems, 2022. xi, 3, 6, 9, 77

[163] A. Saseendran, K. Skubch, and M. Keuper. Multi-class multi-instance count conditioned adversarial image generation. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 6742–6751, 2021. xi, 2, 3, 4, 8, 27

[164] J. Schmidhuber. Deep learning in neural networks: An overview. Neural networks : the official journal of the International Neural Network Society, 61:85–117, 2014. 16

[165] L. Schott, J. Rauber, W. Brendel, and M. Bethge. Towards the first adversarially robust neural network model on mnist. In International Conference on Learning Representations (ICLR), May 2019. 77

[166] Y. Schroecker, M. Vecerík, and J. Scholz. Generative predecessor models for sample-efficient imitation learning. ArXiv, abs/1904.01139, 2019. 21

[167] S. Seguí, O. Pujol, and J. Vitrià. Learning to count with deep object features. In 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 90–96, 2015. 4

[168] J. Selva, A. S. Johansen, S. Escalera, K. Nasrollahi, T. B. Moeslund, and A. Clap'es. Video transformers: A survey. ArXiv, abs/2201.05991, 2022. 21

[169] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L.S. Davis, G. Taylor, and T. Goldstein. Adversarial Training for Free! Curran Associates Inc., Red Hook, NY, USA, 2019. 80

[170] O. Sidla, Y. Lypetskyy, N. Brandle, and S. Seer. Pedestrian detection and tracking for counting applications in crowded situations. Proceedings - IEEE International Conference on Video and Signal Based Surveillance 2006, AVSS 2006, pages 70 – 70, 12 2006. 30

[171] N.V. Smirnov. Approximate distribution laws for random variables, constructed from empirical data" uspekhi mat. Nauk, 10:179–206, 1944. In Russian. 83

[172] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder variational autoencoders. In Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, page 3745–3753, 2016. 18, 19

[173] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. CoRR, abs/1312.6199, 2013. 80

[174] P. Tabacof, J. Tavares, and E. Valle. Adversarial images for variational autoencoders. ArXiv, abs/1612.00155, 2016. 78, 79

[175] Y. Tashiro, J. Song, Y. Song, and S. Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, Advances in Neural Information Processing Systems, 2021. 22

[176] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu. Learning deep representations for graph clustering. In Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014. 53

[177] A. Tirinzoni, R. Poiani, and M. Restelli. Sequential transfer in reinforcement learning with a generative model. In International Conference on Machine Learning, pages 9481–9492. PMLR, 2020. 1

[178] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. Wasserstein auto-encoders. arXiv preprint arXiv:1711.01558, 2017. 13, 18, 19, 52, 62

[179] J. M. Tomczak and M. Welling. Vae with a vampprior. In AISTATS, 2018. 18, 19, 52, 78

[180] A. Touati, H. Satija, J. Romoff, J. Pineau, and P. Vincent. Randomized value functions via multiplicative normalizing flows. In Conference on Uncertainty in Artificial Intelligence, 2018. 21

[181] F. Tramèr and D. Boneh. Adversarial Training and Robustness for Multiple Perturbations. Red Hook, NY, USA, 2019. 80

[182] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. Mcdaniel. Ensemble adversarial training: Attacks and defenses. ArXiv, abs/1705.07204, 2017. 80

[183] D. Tran, K. Vafa, K. K. Agrawal, L. Dinh, and B. Poole. Discrete flows: Invertible generative models of discrete data. In Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, pages 14692–14701, 2019. 21

[184] A. Trask, F. Hill, S. E. Reed, J. W. Rae, C. Dyer, and P. Blunsom. Neural arithmetic logic units. In NeurIPS, 2018. 4, 104

[185] R. E. Turner and M. Sahani. Two problems with variational expectation maximisation for time series models, page 104–124. Cambridge University Press, 2011. 18

[186] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. ArXiv, abs/1609.03499, 2016. 1

[187] A. van den Oord, N. Kalchbrenner, and Koray K. Kavukcuoglu. Pixel recurrent neural networks. In ICML, volume 48, pages 1747–1756, 2016. 11

[188] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30, 2017. 21

[189] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In International Conference on Learning Representations, 2018. 21

[190] P. Verma and C. Chafe. A generative model for raw audio using transformer architectures. 2021 24th International Conference on Digital Audio Effects (DAFx), pages 230–237, 2021. 1

[191] Y. Wang, A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang, F. Zhang, C. Fuegen, G. Zweig, and M. L. Seltzer. Transformer-based acoustic modeling for hybrid speech recognition. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6874–6878, 2019. 21

[192] J. Weston, A. Bordes, S. Chopra, and T. Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. CoRR, abs/1502.05698, 2016. 30

[193] M. Willetts, A. Camuto, T. Rainforth, S Roberts, and C. Holmes. Improving {vae}s' robustness to adversarial attack. In International Conference on Learning Representations, 2021. 18, 78, 79, 85

[194] R. Winterhalder, M. Bellagente, and B. P. Nachman. Latent space refinement for deep generative models. ArXiv, abs/2106.00792, 2021. 1

[195] V. Wolf, A. Lugmayr, M. Danelljan, L. V. Gool, and R. Timofte. Deflow: Learning complex image degradations from unpaired data with conditional flows. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 94–103, 2021. 11, 21

[196] E. Wong, L Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. In International Conference on Learning Representations, 2020. 6, 78, 79, 80, 81, 82

[197] N. Wu, B. Green, X. Ben, and S. O'Banion. Deep transformer models for time series forecasting: The influenza prevalence case. ArXiv, abs/2001.08317, 2020. 21

[198] K. Wynn. Children's understanding of counting. Cognition, 36(2):155–193, 1990. 27, 28

[199] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. ArXiv, abs/1708.07747, 2017. 23, 62

[200] J. Xie, R.Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In International conference on machine learning, pages 478–487, 2016. 53

[201] M. Xu, L. Yu, Y. Song, C. Shi, S. Ermon, and J. Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In International Conference on Learning Representations, 2022. 1, 22

[202] B. Yang, X. Fu, N. D Sidiropoulos, and M. Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 3861–3870. JMLR. org, 2017. 53

[203] Y. Yang, G. Zhang, D. Katabi, and Z. Xu. Me-net: Towards effective adversarial robustness with matrix estimation. In International Conference on Machine Learning, 2019. 80

[204] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. J. Smola. Resnest: Split-attention networks. ArXiv, abs/2004.08955, 2020. 30

[205] M. Zhang, S. Jiang, Z. Cui, R. Garnett, and Y. Chen. D-vae: A variational autoencoder for directed acyclic graphs. In NeurIPS, 2019. 51

[206] S. Zhang, R. Benenson, and B. Schiele. Citypersons: A diverse dataset for pedestrian detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4457–4465, 2017. 37

[207] S. Zhao, J. Song, and S. Ermon. Learning hierarchical features from deep generative models. In Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, page 4091–4099. JMLR.org, 2017. 18, 19

[208] S. Zhao, J. Song, and S. Ermon. Towards deeper understanding of variational autoencoding models. ArXiv, abs/1702.08658, 2017. 18

[209] J. Zhu, T. Park, P. Isola, and A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. pages 2242–2251, 10 2017. 15, 29

[210] Z. M. Ziegler and A. M. Rush. Latent normalizing flows for discrete sequences. ArXiv, abs/1901.10548, 2019. 21

[211] B. Zong, Q. Song, M. Renqiang M., W. Cheng, C. Lumezanu, D. k. Cho, and H. Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In ICLR, 2018. 18, 19