PACGAN: THE POWER OF TWO SAMPLES IN GENERATIVE ADVERSARIAL NETWORKS

BY

ASHISH KUMAR KHETAN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Adviser:

Assistant Professor Sewoong Oh
Assistant Professor Sanmi Koyejo

ABSTRACT

Generative adversarial networks (GANs) are innovative techniques for learning generative models of complex data distributions from samples. Despite remarkable recent improvements in generating realistic images, one of their major shortcomings is the fact that in practice, they tend to produce samples with little diversity, even when trained on diverse datasets. This phenomenon, known as mode collapse, has been the main focus of several recent advances in GANs. Yet there is little understanding of why mode collapse happens and why existing approaches are able to mitigate mode collapse. We propose a principled approach to handling mode collapse, which we call *packing*. The main idea is to modify the discriminator to make decisions based on multiple samples from the same class, either real or artificially generated. We borrow analysis tools from binary hypothesis testing—in particular the seminal result of Blackwell [1]—to prove a fundamental connection between packing and mode collapse. We show that packing naturally penalizes generators with mode collapse, thereby favoring generator distributions with less mode collapse during the training process. Numerical experiments on benchmark datasets suggests that packing provides significant improvements in practice as well.

To my grandmother and parents, for their unconditional love and support.

ACKNOWLEDGMENTS

I wish to express my sincere appreciation and gratitude toward Professor Sewoong Oh for his invaluable guidance and motivation thoughout this thesis work. I also wish to thank my collaborators for this work, Professor Giulia Fanti and fellow graduate student Zinan Lin at CMU. In addition, I wish to thank Professor Sanmi Koyejo for agreeing to be my advisor in the CS department.

Most importantly, I would like to thank my beloved parents. This thesis would not have been complete without their support and patience.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Generative adversarial networks (GANs) are an innovative technique for training generative models to produce realistic examples from a data distribution [3]. Suppose we are given N i.i.d. samples $X_1, \dots, X_N$ from an unknown probability distribution P over some high-dimensional space $\mathbb{R}^p$ (e.g., images). The goal of generative modeling is to learn a model that enables us to produce samples from P that are not in the training data. Classical approaches to this problem typically search over a parametric family (e.g., a Gaussian mixture), and fit parameters to maximize the likelihood of the observed data. Such likelihood-based methods suffer from the curse of dimensionality in real-world datasets, such as images. Deep neural network-based generative models were proposed to cope with this problem [4, 5, 3]. However, these modern generative models can be difficult to train, in large part because it is challenging to evaluate their likelihoods. Generative adversarial networks made a breakthrough in training such models, with an innovative training method that uses a minimax formulation whose solution is approximated by iteratively training two competing neural networks—hence the name "adversarial networks".

GANs have attracted a great deal of interest recently. They are able to *generate* realistic, crisp, and original examples of images [3, 6] and text [7]. This is useful in image and video processing (e.g. frame prediction [8], image super-resolution [9], and image-to-image translation [10]), as well as dialogue systems or chatbots—applications where one may need realistic but artificially generated data. Further, they implicitly learn a *latent, low-dimensional representation* of arbitrary high-dimensional data. Such embeddings have been hugely successful in the area of natural language processing (e.g. word2vec [11]). GANs have the potential to provide such an unsupervised solution to learning representations that capture semantics of the domain to arbitrary data structures and applications.

Primer on GANs. Neural-network-based generative models are trained to map a (typically lower dimensional) random variable $Z \in \mathbb{R}^d$ from a standard distribution (e.g. spherical Gaussian) to a domain of interest, like images. In this context, a *generator* is a function $G : \mathbb{R}^d \to \mathbb{R}^p$, which is chosen from a rich class of parametric functions like deep neural networks. In unsupervised generative modeling, one of the goals is to train the parameters of such a generator from unlabelled training data drawn independently from some real world dataset (such as celebrity faces in CelebA [12] or natural images from CIFAR-100 [13]), in order to produce examples that are realistic but different from the training data.

A breakthrough in training such generative models was achieved by the innovative idea of

GANs [3]. GANs train two neural networks: one for the generator $G(Z)$ and the other for a discriminator $D(X)$. These two neural networks play a dynamic minimax game against each other. An analogy provides the intuition behind this idea. The generator is acting as a forger trying to make fake coins (i.e., samples), and the discriminator is trying to detect which coins are fake and which are real. If these two parties are allowed to play against each other long enough, eventually both will become good. In particular, the generator will learn to produce coins that are indistinguishable from real coins (but preferably different from the training coins he was given).

Concretely, we search for (the parameters of) neural networks G and D that optimize the following type of minimax objective:

$$\begin{aligned} G^* &\in \arg\min_{G} \max_{D} V(G, D) \\ &= \arg\min_{G} \max_{D} \mathbb{E}_{X\sim P}[\log(D(X))] + \mathbb{E}_{Z\sim P_Z}[\log(1 - D(G(Z)))] , \end{aligned} \tag{1.1}$$

where P is the distribution of the real data, and P_Z is the distribution of the input code vector Z. Here D is a function that tries to distinguish between real data and generated samples, whereas G is the mapping from the latent space to the data space. Critically, [3] shows that the global optimum of (1.1) is achieved if and only if $P = Q$, where Q is the generated distribution of $G(Z)$. We refer to Chapter 5 for a detailed discussion of this minimax formulation. The solution to the minimax problem (1.1) can be approximated by iteratively training two "competing" neural networks, the generator G and discriminator D. Each model can be updated individually by backpropagating the gradient of the loss function to each model's parameters.

Mode Collapse in GANs. One major challenge in training GAN is a phenomenon known as *mode collapse*, which collectively refers to the lack of diversity in generated samples. One manifestation of mode collapse is the observation that GANs commonly miss some of the modes when trained on multimodal distributions. For instance, when trained on hand-written digits with ten modes, the generator might fail to produce some of the digits [14]. Similarly, in tasks that translate a caption into an image, generators have been shown to generate series of nearly-identical images [15]. Mode collapse is believed to be related to the training instability of GANs—another major challenge in GANs.

Several approaches have been proposed to fight mode collapse, e.g. [16, 17, 2, 14, 18, 19, 20, 21]. We discuss prior work on mode collapse in detail in Chapter 2. Proposed solutions rely on modified architectures [16, 17, 2, 14], loss functions [19, 22], and optimization algorithms [18]. Although each of these proposed methods is empirically shown to help mitigate mode collapse, it is not well understood how the proposed changes relate to mode collapse. Previously-proposed heuristics fall short of providing rigorous explanations on why they achieve empirical gains, especially when those gains are sensitive to architecture hyperparameters.

Our Contributions. In this work, we examine GANs through the lens of *binary hypothesis testing*. By viewing the discriminator as performing a binary hypothesis test on samples (i.e., whether they were drawn from distribution P or Q), we can apply insights from classical hypothesis testing literature to the analysis of GANs. In particular, this hypothesis-testing viewpoint provides a fresh perspective and understanding of GANs that leads to the following contributions:

1. The first contribution is conceptual: we propose a formal mathematical definition of mode collapse that abstracts away the geometric properties of the underlying data distributions (see Chapter 5.1). This definition is closely related to the notions of false alarm and missed detection in binary hypothesis testing (see Chapter 5.2.1). Given this definition, we provide a new interpretation of the pair of distributions (P, Q) as a two-dimensional region called the *mode collapse region*, where P is the true data distribution and Q the generated one. The mode collapse region provides new insights on how to reason about the relationship between those two distributions (see Chapter 5.1).

2. The second contribution is analytical: through the lens of hypothesis testing and mode collapse regions, we show that if the discriminator is allowed to see samples from the m-th order product distributions P^m and Q^m instead of the usual target distribution P and generator distribution Q, then the corresponding loss when training the generator naturally penalizes generator distributions with strong mode collapse (see Chapter 5.2). Hence, a generator trained with this type of discriminator will be encouraged to choose a distribution that exhibits less mode collapse. The *region* interpretation of mode collapse and corresponding data processing inequalities provide the analysis tools that allows us to prove strong and sharp results with simple proofs (see Chapter 6). This follows a long tradition in information theory literature (e.g. [23, 24, 25, 26, 27, 28, 29, 30, 31]) where operational interpretations of mutual information and corresponding data processing inequalities have given rise to simple proofs of strong technical results.

3. The third contribution is algorithmic: based on the insights from the region interpretation of mode collapse, we propose a new GAN framework to mitigate mode collapse,

which we call PacGAN. PacGAN can be applied to any existing GAN, and it requires only a small modification to the discriminator architecture (see Chapter 3). The key idea is to pass m "packed" or concatenated samples to the discriminator, which are jointly classified as either real or generated. This allows the discriminator to do binary hypothesis testing based on the product distributions (P^m, Q^m), which naturally penalizes mode collapse (as we show in Chapter 5.2). We demonstrate on benchmark datasets that PacGAN significantly improves upon competing approaches in mitigating mode collapse (see Chapter 4). Further, unlike existing approaches on jointly using multiple samples, e.g. [14], PacGAN requires no hyper parameter tuning and incurs only a slight overhead in the architecture.

Outline. This work is structured as follows: first we describe in detail the related work on GANs in general and mode collapse in particular in Chapter 2. In Chapter 3, we present the PacGAN framework , and evaluate it empirically according to the metrics and experiments proposed in prior work (Chapter 4). In Chapter 5, we propose a *new* definition of mode collapse, and provide analyses showing that PacGAN mitigates mode collapse. The proofs of the main results are provided in Chapter 6.

CHAPTER 2: RELATED WORK

The literature on GANs has documented three primary, closely-related challenges: (*i*) they are unstable to train, (*ii*) they are challenging to evaluate, and (*iii*) they exhibit mode collapse (more broadly, they do not generalize). Much research has emerged in recent years addressing these challenges. Our work explicitly addresses the challenge (*iii*). We give a brief overview of the related work on each of these challenges, and its relation to our work.

Training instability. GANs' alternating generator and discriminator updates can lead to significant instability during training. This instability manifests itself as oscillating values of the loss function that exceed variations caused by minibatch processing [32]. Such variability makes it challenging to evaluate when training has converged, let alone which model one should choose among those obtained throughout the training process. This phenomenon is believed to arise because in practice, the learned distribution and the true distribution lie on disjoint manifolds in a high-dimensional space [22]. As such, the discriminator can often learn to perfectly distinguish generated and real samples. On real data, the discriminator (correctly) learns to output '1', and vice versa on generated data. This is believed in GAN literature to cause the generator loss function to have a negligible gradient, leading to unstable parameter updates.

Several papers have proposed methods for mitigating this instability, generally taking one of two approaches. The first relies on changing the optimized distance metric. Regular GANs optimize the Jensen-Shannon divergence between the true distribution and the learned one [3]. Jensen-Shannon divergence can behave poorly in regions where the two distributions have nonoverlapping support [22], so other works have proposed alternative distance metrics, including Wasserstein distance [22] and neural network distance [33].

Another approach is to propose architectural changes that empirically improve training stability. For example, Salimans et al. proposed a number of heuristic tricks for improving the training of GANs, including minibatch discrimination, reference batch normalization, and feature mapping [14]. Our work most closely resembles minibatch discrimination from [14], which also inputs multiple images to the discriminator. We provide a detailed comparison between this proposed minibatch discriminator and ours later in this Chapter.

Evaluation Techniques. Generative models (including GANs) are notoriously difficult to evaluate. Ideally, one would measure the distance between the true distribution and the learned one. However, typical generative models can only produce *samples* from a learned distribution, and on real datasets, the true distribution is often unknown. As such, prior work on GANs has used a number of heuristic evaluation techniques.

The most common evaluation technique is visual inspection. Many papers produce a collection of generated images, and compare them to the underlying dataset [3, 34, 32], or ask annotators to evaluate the realism of generated images [14]. This approach can be augmented by interpolating between two points in the latent space and illustrating that the GAN produces a semantically meaningful interpolation between the generated images [17]. This approach is useful to the extent that some GANs produce visually unrealistic images, but it is expensive, unreliable, and it does not help identify generalization problems [35].

Another common approach involves estimating the likelihood of a holdout set of test data under the learned distributions. The learned distribution is estimated using a standard kernel density estimator (KDE)[36]. However, KDEs are known to have poor performance in high dimensions, and in practice, the error in KDE is often larger than the distance between real and learned distributions [36]. Hence, it is unclear how meaningful such estimates are. One proposed approach uses annealed importance sampling (AIS) instead of KDE to estimate log-likelihoods [36], with significantly increased accuracy levels.

An increasing number of papers are using *classification-based evaluation metrics*. Naively, GANs trained on labelled datasets can pass their outputs through a pre-trained classifier. The classifier outputs indicate which modes are represented in the generated samples [17, 14, 2]. This is useful for measuring the first type of mode collapse (missing modes), but it cannot reveal the second type (partial collapse within a mode). To provide a more nuanced view of the problem, [37] recently proposed a more general classification-based evaluation metric, in which they train a classifier on generated data and real data, and observe differences in classifier performance on a holdout set of test data. While this approach does not directly evaluate partial mode collapse, it is more likely to implicitly measure it by producing weaker classifiers when trained on generated data. On datasets that are not labelled, some papers have relied on *human* classification, asking human annotators to 'discriminate' whether an image is real or generated [6].

Mode Collapse/Generalization. Mode collapse collectively refer to the phenomenon of lack of divergence in the generated samples. This includes trained generators assigning low probability mass to significant subsets of the data distribution's support, and hence losing some modes. This also includes the phenomenon of trained generators mapping two latent vectors that are far apart to the same or similar data samples. Mode collapse is a byproduct of poor generalization—i.e., the generator does not learn the true data distribution; this phenomenon is a topic of recent interest [33, 38]. Prior work has observed two types of mode collapse: entire modes from the input data are missing from the generated data (e.g., in a dataset of animal pictures, lizards never appear), or the generator only creates images within a subset of a particular mode (e.g., lizards appear, but only lizards that are a particular shade of green) [32, 39, 38, 16, 18, 15]. These phenomena are not well-understood, but a number of explanatory hypotheses have been proposed:

1. The objective function is ill-suited to the problem [22], potentially causing distributions that exhibit mode collapse to be local minima in the optimization objective function.

2. Weak discriminators cannot detect mode collapse, either due to low capacity or a poorly-chosen architecture [18, 14, 33, 40].

3. The maximin solution to the GAN game is not the same as the minimax solution [32].

The impact and interactions of these hypotheses are not well-understood, but we show in this work that a packed discriminator can significantly reduce mode collapse, both theoretically and in practice. In particular, the method of packing is simple, and leads to clean theoretical analyses. We compare the proposed approach of packing to three main approaches in the literature for mitigating mode collapse:

(1) *Joint Architectures.* The most common approach to address mode collapse involves an encoder-decoder architecture, in which the GAN learns an encoding $G^{-1}(X)$ from the data space to a lower-dimensional latent space, on top of the usual decoding $G(Z)$ from the latent space to the data space. Examples include bidirectional GANs [17], adversarially learned inference (ALI) [16], and VEEGAN [2]. These joint architectures feed both the latent and the high-dimensional representation of each data point into the discriminator: $\{(Z_i, G(Z_i))\}$ for the generated data and $\{(G^{-1}(X_i), X_i)\}$ for the real data. In contrast, classical GANs use only the decoder, and feed only high-dimensional representations into the discriminator. Empirically, training these components jointly seems to improve the GAN performance overall, while also producing useful feature vectors that can be fed into downstream tasks like classification. Nonetheless, we find experimentally that using the same generator architectures and discriminator architectures, packing captures more modes than these joint architectures, with significantly less overhead in the architecture and computation.

(2) *Augmented Discriminators.* Several papers have observed that discriminators lose discriminative power by observing only one (unlabeled) data sample at a time [32, 14]. A natural solution for labelled datasets is to provide the discriminator with image labels. This has been found to work well in practice [19], though it does not generalize to unlabelled data. A more general technique is *minibatch discrimination* [14]. Like our proposed packing architecture, minibatch discrimination feeds an array of data samples to the discriminator.

However, unlike packing, minibatch discrimination proposed in [14] is complicated both computationally and conceptually, and highly sensitive to the delicate hyper-parameter choices. At a high level, the main idea in minibatch discrimination is to give the discriminator a side information coming from a minibatch, and use it together with each of the single example in the minibatch to classify each sample. The proposed complex architecture to achieve this goal is as follows.

Let $f(X_i)$ denote a vector of (latent) features for input X_i produced by some intermediate layer in the discriminator. A tensor T is learned such that the tensor product $T[\mathbb{I}, \mathbb{I}, f(X_i)]$ gives a latent matrix representation M_i of the input X_i. The notation $T[\mathbb{I}, \mathbb{I}, f(X_i)]$ indicates a tensor to matrix linear mapping, where you take the third dimension and apply a vector $f(X_i)$. The L_1 distance across the rows of the M_i's are computed for each pair of latent matrices in the minibatch to give a measure $c_b(X_i, X_j) = \exp(-\|M_{i,b} - M_{j,b}\|_{L_1})$. This minibatch layer outputs $o(X_i)_b = \sum_{j=1}^{n} c_b(X_i, X_j)$. This is concatenated with the original latent feature $f(X_i)$ to be passed through the upper layers of the discriminator architecture. While the two approaches start from a similar intuition that batching or packing multiple samples gives stronger discriminator, the proposed architectures are completely different. PacGAN is easier to implement, quantitatively shows strong performance in experiments, and is principled: our theoretical analysis rigorously shows that packing is a principled way to use multiple samples at the discriminator.

(3) *Optimization-based solutions.* Another potential source of mode collapse is imperfect optimization algorithms. Exact optimization of the GAN minimax objective function is computationally intractable, so GANs typically use iterative parameter updates between the generator and discriminator: for instance, we update the generator parameters through k_1 gradient descent steps, followed by k_2 discriminator parameter updates. Recent work has studied the effects of this compromise, showing that iterative updates can lead to non-convergence in certain settings [40]—a worse problem than mode collapse. Unrolled GANs [18] propose a middle ground, in which the optimization takes k (usually five) gradient steps into account when computing gradients. These unrolled gradients affect the generator parameter updates by better predicting how the discriminator will respond. This approach is conjectured to spread out the generated samples, making it harder for the discriminator to distinguish real and generated data. The primary drawback of this approach is computational cost; packing achieves better empirical performance with smaller computational overhead and training complexity.

Theoretical results on GANs. A breakthrough in theoretical analysis of GANs was achieved by Arora et al. in [33], where several theoretical contributions were made. Recall that typical assumption in theoretical analyses is (a) infinite samples that allows one to work with population expectations, and (b) infinite expressive power at the discriminator. This seminal work addresses both of these assumptions in the following way. First, to show that existing losses (such as Wasserstein loss [22] and cross entropy loss [3]) do not generalize, [33] relaxes both (a) and (b). Under this quite general setting, a GAN is trained on these typical choices of losses with a target distribution of a spherical Gaussian. Then, with a discriminator with enough expressive power, the training loss will converge to its maximum, which is proven to be strictly bounded away from zero for this Gaussian example. The implication of this analysis is that a perfect generator with infinite expressive power still will not be able to generate the target Gaussian distribution, as it is penalize severely in the empirical loss defined by the training samples. This observation leads to the second contribution of the work, where a proper distance is defined, called *neural network divergence*, that takes into account the finite expressive power of neural networks. It is proven that the neural network divergence has a much better generalization properties than Jensen-Shannon divergence or Wasserstein distance. This implies that this new neural network distance can better capture how the GAN performs for a specific choice of a loss. Based on this intuition, a new class of generators called, MIX+GAN, are proposed that are provably shown to fool a neural network discriminator with finite expressive power (i.e. finite number of parameters).

Liu et al. study the effects of discriminator family with finite expressive power and the distributional convergence properties of various choices of the loss functions in [41]. It is shown that the restricted expressive power of the discriminator (including the popular neural network based discriminators) have the effect of encouraging moment-matching conditions to be satisfied. Further, it is shown that for a broad class of loss functions, convergence in the loss function implies distributional weak convergence, which generalizes known convergence results of [42, 22].

In [40], Li et al. take a first step towards understanding of GAN training dynamics. A particular training example of learning from a mixture of two Gaussians is carefully studied, which exhibit all all common failure cases: mode collapse and oscillatory behavior. This holds even with improve GAN training such as unrolled GANs [18]. However, it is both experimentally and theoretically shown that GAN with an optimal discriminator converges. This is first convergence proof of a non-trivial GAN dynamics, and shows a clear dichotomy between the GAN dynamics from an optimal discriminator and one from a more practical simultaneous updates. This analyses reveal the root cause, called *discriminator collapse*; when the generator is good at fooling the discriminator, the discriminator gradient updates are stuck in a local minimum.

Feizi et al. address the effect of generator and discriminator architectures for a simpler case of learning a single Gaussian distribution in [43]. By connecting the loss function to supervised learning, the generalization performance of a simple LQG-GAN is analyzed where the generator is linear, the loss is quadratic, and the data is coming from a Gaussian distribution. An interesting connection between principal component analysis and the optimal generator of this particular GAN is made. The sample complexity of this problem is shown to be linear in the dimension, if the discriminator is constrained to be quadratic, where as for general discriminators the sample complexity can be much larger.

CHAPTER 3: PACGAN: A NOVEL FRAMEWORK FOR MITIGATING MODE COLLAPSE

We propose a new framework for mitigating mode collapse in GANs. We start with an arbitrary existing GAN[1], which is typically defined by a generator architecture, a discriminator architecture, and a loss function. Let us call this triplet the *mother architecture*.

The PacGAN framework maintains the same generator architecture and loss function as the mother architecture, and makes a slight change only to the discriminator. That is, instead of using a discriminator $D(X)$ that maps a single (either from real data or from the generator) to a (soft) label, we use an *augmented* discriminator $D(X_1, X_2, \ldots, X_m)$ that maps m samples, jointly coming from either real data or the generator, to a single (soft) label. These m samples are drawn independently from the same distribution—either real (jointly labelled as $Y = 1$) or generated (jointly labelled as $Y = 0$). We refer to the concatenation of samples with the same label as *packing*, the resulting concatenated discriminator as a *packed discriminator*, and the number m of concatenated samples as the *degree of packing*. We call this approach a framework instead of an architecture, because the proposed approach of packing can be applied to any existing GAN, using any architecture and any loss function, as long as it uses a discriminator of the form $D(X)$ that classifies a single input sample.

We propose the nomenclature "Pac(X)(m)" where (X) is the name of the mother architecture, and (m) is an integer that refers to how many samples are packed together as an input to the discriminator. For example, if we take an original GAN and feed the discriminator three packed samples as input, we call this "PacGAN3". If we take the celebrated DCGAN [34] and feed the discriminator four packed samples as input, we call this "PacDCGAN4". When we refer to the generic principle of packing, we use PacGAN without an subsequent integer.

How to pack a discriminator. Note that there are many ways to change the discriminator architecture to accept packed input samples. We propose to keep all hidden layers of the discriminator exactly the same as the mother architecture, and only increase the number of nodes in the input layer by a factor of m. For example, in Figure 3.1, suppose we start with a mother architecture in which the discriminator is a fully-connected feed-forward network. Here, each sample X lies in a space of dimension $p = 2$, so the input layer has two nodes. Now, under PacGAN2, we would multiply the size of the input layer by the packing degree (in this case, two), and the connections to the first hidden layer would be adjusted so that the first two layers remain fully-connected, as in the mother architecture. The grid-patterned

[1] For a list of some popular GANs, we refer to the GAN zoo: https://github.com/hindupuravinash/the-gan-zoo

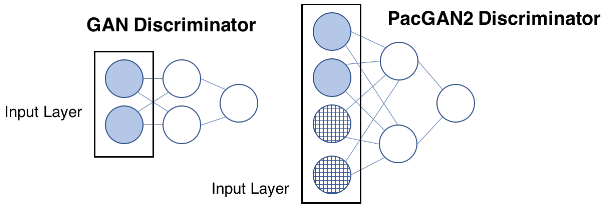


Figure 3.1: PacGAN(m) augments the input layer by a factor of m. The number of edges between the first two layers are increased accordingly to preserve the connectivity of the mother architecture (typically fully-connected). Packed samples are fed to the input layer in a concatenated fashion; the grid-patterned nodes represent input nodes for the second input sample.

nodes in Figure 3.1 represent input nodes for the second sample.

Similarly, when packing a DCGAN, which uses convolutional neural networks for both the generator and the discriminator, we simply stack the images into a tensor of depth m. For instance, the discriminator for PacDCGAN5 on the MNIST dataset of handwritten images [44] would take an input of size $28 \times 28 \times 5$, since each individual black-and-white MNIST image is 28×28 pixels. Only the input layer and the number of weights in the corresponding first convolutional layer will increase in depth by a factor of five. By modifying only the input dimension and fixing the number of hidden and output nodes in the discriminator, we can focus purely on the effects of *packing* in our numerical experiments in Chapter 4.

How to train a packed discriminator. Just as in standard GANs, we train the packed discriminator with a bag of samples from the real data and the generator. However, each minibatch in the stochastic gradient descent now consists of *packed* samples. Each packed sample is of the form $(X_1, X_2, \ldots, X_m, Y)$, where the label is $Y = 1$ for real data and $Y = 0$ for generated data, and the m independent samples from either class are jointly treated as a single, higher-dimensional feature $(X_1, \ldots, X_m)$. The discriminator learns to classify m packed samples jointly. Intuitively, packing helps the discriminator detect mode collapse because lack of diversity is more obvious in a set of samples than in a single sample. Fundamentally, packing allows the discriminator to observe samples from *product distributions*, which highlight mode collapse more clearly than unmodified data and generator distributions. We make this statement precise in Chapter 5.

Notice that the computational overhead of PacGAN training is marginal, since only the input layer of the discriminator gains new parameters. Furthermore, we keep all training hyperparameters identical to the mother architecture, including the stochastic gradient descent minibatch size, weight decay, learning rate, and the number of training epochs. This is in contrast with other approaches for mitigating mode collapse that require significant computational overhead and/or delicate hyperparameter selection [17, 16, 14, 2, 18].

Computational complexity. The exact computational complexity overhead of PacGAN (compared to GANs) is architecture-dependent, but can be computed in a straightforward manner. For example, consider a discriminator with w fully-connected layers, each containing g nodes. Since the discriminator has a binary output, the $(w+1)$th layer has a single node, and is fully connected to the previous layer. We seek the computational complexity of a single minibatch parameter update, where each minibatch contains r samples. Backpropagation in such a network is dominated by the matrix-vector multiplication in each hidden layer, which has complexity $O(g^2)$ per input sample, assuming a naive implementation. Hence the overall minibatch update complexity is $O(rwg^2)$. Now suppose the input layer is expanded by a factor of m. If we keep the same number of minibatch elements, the per-minibatch cost grows to $O((w+m)rg^2)$. We find that in practice, even $m = 2$ or $m = 3$ give good results.

CHAPTER 4: EXPERIMENTS

On standard benchmark datasets, we compare PacGAN to several baseline GAN architectures, some of which are explicitly proposed to mitigate mode collapse: GAN [3], DCGAN [14], VEEGAN [2], Unrolled GANs [18], and ALI [17]. We also implicitly compare against BIGAN [16], which is conceptually identical to ALI. To isolate the effects of packing, we make minimal choices in the architecture and hyperparameters of our packing implementation. For each experiment, we evaluate packing by taking a standard, baseline GAN implementation that was *not* designed to prevent mode collapse, and adding packing in the discriminator. In particular, our goal for this Chapter is to reproduce experiments from existing literature, apply the packing framework to the simplest GAN among those in the baseline, and showcase how packing affects the performance.

Metrics. For consistency with prior work, we measure several previously-used metrics. On datasets with clear, known modes (e.g., Gaussian mixtures, labelled datasets), prior papers have counted the *number of modes* that are produced by a generator [16, 18, 2]. In labelled datasets, this number can be evaluated using a third-party trained classifier that classifies the generated samples [14]. In Gaussian Mixture Models (GMMs), for example in [2], a mode is considered lost if there is no sample in the generated test data within x standard deviations from the center of that mode. In [2], x is set to be three for 2D-ring and 2D-grid, and ten for 1200D-synthetic. A second metric used in [2] is the *number of high-quality samples*, which is the proportion of the samples that are within x standard deviation from the center of a mode. Finally, the *reverse Kullback-Leibler divergence* over the modes has been used to measure the quality of mode collapse as follows. Each of the generated test samples is assigned to its closest mode; this induces an empirical, discrete distribution with an alphabet size equal to the number of observed modes in the generated samples. A similar induced discrete distribution is computed from the real data samples. The reverse KL divergence between the induced distribution from generated samples and the induced distribution from the real samples is used as a metric. Each of these three metrics has shortcomings—for example, the number of observed modes does not account for class imbalance among generated modes, and all of these metrics only work for datasets with known modes. Defining an appropriate metric for evaluating GANs is an active research topic [35, 36, 37].

Datasets. We use a number of synthetic and real datasets for our experiments, all of which have been studied or proposed in prior work. The **2D-ring** [2] is a mixture of eight two-dimensional spherical Gaussians with means $(\cos((2\pi/8)i), \sin((2\pi/8)i))$ and variances 10^{-4}

in each dimension for $i \in \{1, \ldots, 8\}$. The **2D-grid** [2] is a mixture of 25 two-dimensional spherical Gaussians with means $(-4 + 2i, -4 + 2j)$ and variances 0.0025 in each dimension for $i, j \in \{0, 1, 2, 3, 4\}$.

To examine real data, we use the MNIST dataset [44], which consists of 70,000 images of handwritten digits, each 28×28 pixels. Unmodified, this dataset has 10 modes, one for each digit. As done in Mode-regularized GANs [19], Unrolled GANs [18] and VEEGAN [2], we augment the number of modes by *stacking* the images. That is, we generate a new dataset of 128,000 images, in which each image consists of three randomly-selected MNIST images that are stacked into a $28 \times 28 \times 3$ image in RGB. This new dataset has (with high probability) $1000 = 10 \times 10 \times 10$ modes. We refer to this as the **stacked MNIST** dataset.

4.1 SYNTHETIC DATA EXPERIMENTS FROM VEEGAN [2]

Our first experiment evaluates the number of modes and the number of high-quality samples for the 2D-ring and the 2D-grid. Results are reported in Table 4.1. The first four rows are copied directly from Table 1 in [2]. The last three rows contain our own implementation of PacGANs. We do not make any choices in the hyper-parameters, the generator architecture, the discriminator architecture, and the loss. Our implementation attempts to reproduce the VEEGAN architecture to the best of our knowledge, as described below.

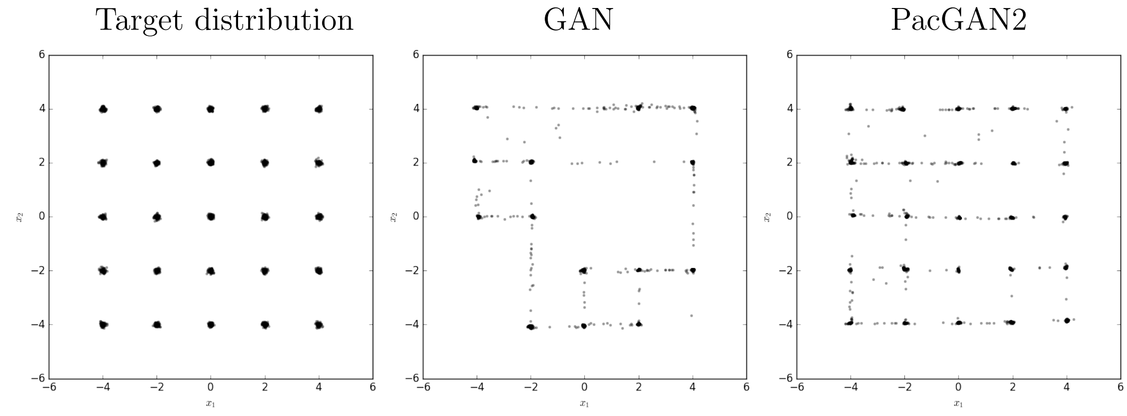


Figure 4.1: Scatter plot of the 2D samples from the true distribution (left) of 2D-grid and the learned generators using GAN (middle) and PacGAN2 (right). PacGAN2 captures all of the 25 modes.

Architecture and hyper-parameters. All of the GANs we implemented in this experiment use the same overall architecture, which is chosen to match the architecture in

VEEGAN's code [2]. The generators have two hidden layers, 128 units per layer with ReLU activation, trained with batch normalization [45]. The input noise is a two dimensional spherical Gaussian with zero mean and unit variance. The discriminator has one hidden layer, 128 units on that layer. The hidden layer uses LinearMaxout with 5 maxout pieces, and no batch normalization is used in the discriminator.

We train each GAN with 100,000 total samples, and a mini-batch size of 100 samples; training is run for 200 epochs. The discriminator's loss function is $\log(1+\exp(-D(\text{real data})))+\log(1+\exp(D(\text{generated data})))$, except for VEEGAN which has an additional regularization term. The generator's loss function is

$$\log(1+\exp(D(\text{real data})))+\log(1+\exp(-D(\text{generated data}))).$$

Adam [46] stochastic gradient descent is applied with the generator weights and the discriminator weights updated once per mini-batch. At testing, we use 2500 samples from the learned generator for evaluation. Each metric is evaluated and averaged over 10 trials.

	2D-ring		2D-grid	
	Modes (Max 8)	high quality samples	Modes (Max 25)	high quality samples
GAN [3]	1.0	99.30 %	3.3	0.5 %
ALI [17]	2.8	0.13 %	15.8	1.6 %
Unrolled GAN [18]	7.6	35.60 %	23.6	16.0 %
VEEGAN [2]	8.0	52.90 %	24.6	40.0 %
PacGAN2 (ours)	8.0±0.0	78.5±7.7 %	24.6±0.9	65.8±13.4 %
PacGAN3 (ours)	8.0±0.0	84.0±6.1 %	24.9±0.3	71.4±13.8 %
PacGAN4 (ours)	8.0±0.0	82.7±11.3 %	25.0±0.0	76.0±7.1 %

Table 4.1: Two measures of mode collapse proposed in [2] for two synthetic mixtures of Gaussians: number of modes captured by the generator and percentage of high quality samples. Our results are averaged over 10 trials shown with the standard error.

Results. Table 4.1 shows that PacGAN outperforms or matches the baseline schemes, both in the number of modes captured and the percentage of high quality samples. As expected, increasing the degree m of packing seems to increase the average number of modes found, though the increases are marginal for easy tasks. In the 2D grid and ring, we find that PacGAN slightly outperforms VEEGAN, but the proposed datasets seem not to be challenging enough to highlight meaningful differences. However, one can clearly see the gain of packing by comparing the GAN in the first row (which is the mother architecture) and PacGANs in the last rows. The simple change we make to the mother architecture according to the principle of packing makes a significant difference in performance, and the overhead of changes made to the mother architecture are minimal compared to the baselines [17, 18, 2].

Note that maximizing the number of high-quality samples is not necessarily indicative of a good generative model. First, we expect some fraction of probability mass to lie outside the "high-quality" boundary, and that fraction increases with the dimensionality of the dataset. For reference, we find empirically that the expected fraction of high-quality samples in the true data distribution for the 2D ring and grid are both 98.9%, which corresponds to the theoretical ratio for a single 2D Gaussian. These values are higher than the fractions found by PacGAN, indicating room for improvement. However, a generative model could output 100% high-quality points by learning very few modes (as is the case for GANs in the 2D ring in Table 4.1).

Note that our goal is not to compete with the baselines of ALI, Unrolled GAN, and VEEGAN, but to showcase the improvement that can be obtained with packing. In this spirit, we can easily apply our framework to other baselines and test "PacALI", "PacUnrolledGAN", and "PacVEEGAN". In fact, we expect that most GAN architectures can be packed to improve sample quality. However, for these benchmark tests, we see that packing the simplest GAN is sufficient.

4.2 STACKED MNIST EXPERIMENTS

In our next experiments, we evaluate mode collapse on the stacked MNIST dataset (described at the beginning of Chapter 4). These experiments are direct comparisons to analogous experiments in VEEGAN [2] and Unrolled GANs [18]. For these evaluations, we generate 26,000 samples from the generator. Each of the three channels in each sample is classified by a pre-trained third-party MNIST classifier, and the resulting three digits determine which of the $1,000$ modes the sample belongs to. We measure the number of modes captured, as well as the KL divergence between the generated distribution over modes and the expected true one (i.e., a uniform distribution over the 1,000 modes).

Hyperparameters. For these experiments, we train each GAN on 128,000 samples, with a mini-batch size of 64. The generator's loss function is -log(D(generated data)), and the discriminator's loss function is -log(D(real data))-log(1-D(generated data)). We update the generator parameters twice and the discriminator parameters once in each mini-batch, and train the generators over 50 epochs. For testing, we generate 26,000 samples, and evaluate

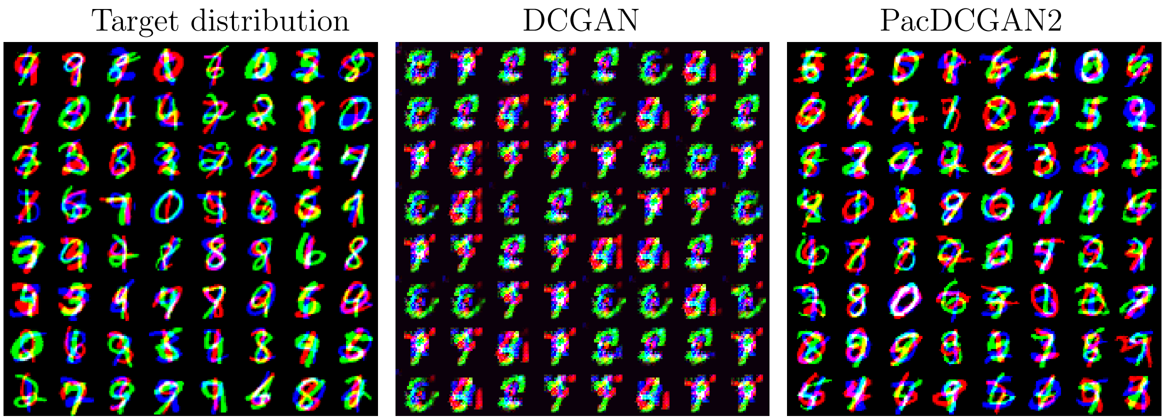


Figure 4.2: True distribution (left), DCGAN generated samples (middle), and PacDCGAN2 generated samples (right) from the stacked-MNIST dataset show PacDCGAN2 captures more diversity while producing sharper images.

the empirical KL divergence and number of modes covered. Finally, we average these values over 10 runs of the entire pipeline.

4.2.1 VEEGAN [2] Experiment

In this experiment, we replicate Table 2 from [2], which measured the number of observed modes in a generator trained on the stacked MNIST dataset, as well as the KL divergence of the generated mode distribution.

Architecture. In line with prior work [2], we used a DCGAN-like architecture for these experiments, which is based on the code at `https://github.com/carpedm20/DCGAN-tensorflow`. In particular, the generator and discriminator architectures are as follows:

Generator:

layer	#outputs	kernel size	stride	activation
Input: $z \sim U(-1,1)^{100}$	100			
Fully connected	2*2*512			ReLU
Transposed Convolution	4*4*256	5*5	2	ReLU
Transposed Convolution	7*7*128	5*5	2	ReLU
Transposed Convolution	14*14*64	5*5	2	ReLU
Transposed Convolution	28*28*3	5*5	2	Tanh

Discriminator (for PacDCGANm):

layer	#outputs	kernel size	stride	BN	activation
Input: $x \sim p_{data}^m$	28*28*(3*m)				
Convolution	14*14*64	5*5	2		LeakyReLU
Convolution	7*7*128	5*5	2	Yes	LeakyReLU
Convolution	4*4*256	5*5	2	Yes	LeakyReLU
Convolution	2*2*512	5*5	2	Yes	LeakyReLU
Fully connected	1				Sigmoid

Results. Results are shown in Table 4.2. Again, the first four rows are copied directly from [2]. The last three rows are computed using a basic DCGAN, with packing in the discriminator. We find that packing gives good mode coverage, reaching all 1,000 modes in every trial. Given a DCGAN that can capture at most 99 modes on average (our mother architecture), the principle of packing, which is a small change in the architecture, is able to improve performance to capture all 1,000 modes. Again we see that packing the simplest DCGAN is sufficient to fully capture all the modes in this benchmark tests, and we do not pursue packing more complex baseline architectures. Existing approaches to mitigate mode collapse, such as ALI, Unrolled GANs, and VEEGAN, are not able to capture as many modes.

	Stacked MNIST	
	Modes (Max 1000)	KL
DCGAN [34]	99.0	3.40
ALI [17]	16.0	5.40
Unrolled GAN [18]	48.7	4.32
VEEGAN [2]	150.0	2.95
PacDCGAN2 (ours)	1000.0±0.0	0.06±0.01
PacDCGAN3 (ours)	1000.0±0.0	0.06±0.01
PacDCGAN4 (ours)	1000.0±0.0	0.07±0.01

Table 4.2: Two measures of mode collapse proposed in [2] for the stacked MNIST dataset: number of modes captured by the generator and reverse KL divergence over the generated mode distribution.

Note that other classes of GANs may also be able to learn most or all of the modes if tuned properly. For example, [18] reports that regular GANs can learn all 1,000 modes even without unrolling if the discriminator is large enough, and if the discriminator is half the size of the generator, unrolled GANs recover up to 82% of the modes when the unrolling parameter is increased to 10. To explore this effect, we conduct further experiments on unrolled GANs in Chapter 4.2.2.

4.2.2 Unrolled GAN [18] Experiment

This experiment is designed to replicate Table 1 from Unrolled GANs [18]. Unrolled GANs exploit the observation that iteratively updating discriminator and generator model parameters can contribute to training instability. To mitigate this, they update model parameters by computing the loss function's gradient with respect to $k \geq 1$ sequential discriminator updates, where k is called the unrolling parameter. [18] reports that unrolling improves mode collapse as k increases, at the expense of greater training complexity.

Unlike Chapter 4.2.1, which reported a single metric for unrolled GANs, this experiment studies the effect of the unrolling parameter and the discriminator size on the number of modes learned by a generator. The key differences between these trials and the unrolled GAN row in Table 4.2 are threefold: (1) the unrolling parameters are different, (2) the discriminator sizes are different, and (3) the generator and discriminator architectures are chosen according to Appendix E in [18].

Results. Our results are reported in Tables 4.3 and 4.4. The first four rows are copied from [18]. As before, we find that packing seems to increase the number of modes covered. Additionally, in both experiments, PacDCGAN finds more modes on average than Unrolled GANs with $k = 10$, with lower reverse KL divergences between the mode distributions. This suggests that packing has a more pronounced effect than unrolling. However, note that the standard error for PacDCGANs is larger than that reported in [18]; this may be due to our relatively small sample size of 10.

	D is 1/4 size of G	
	Modes (Max 1000)	KL
DCGAN [34]	30.6±20.73	5.99±0.42
Unrolled GAN, 1 step [18]	65.4±34.75	5.91±0.14
Unrolled GAN, 5 steps [18]	236.4±63.30	4.67±0.43
Unrolled GAN, 10 steps [18]	327.2±74.67	4.66±0.46
PacDCGAN2 (ours)	370.8±244.34	3.33±1.02
PacDCGAN3 (ours)	534.3±103.68	2.11±0.52
PacDCGAN4 (ours)	557.7±101.37	2.06±0.61

Table 4.3: Modes covered and KL divergence for unrolled GANs as compared to PacDCGANs for various unrolling parameters, discriminator sizes, and the degree of packing.

	D is 1/2 size of G	
	Modes (Max 1000)	KL
DCGAN [34]	628.0±140.9	2.58±0.75
Unrolled GAN, 1 step [18]	523.6±55.77	2.44±0.26
Unrolled GAN, 5 steps [18]	732.0±44.98	1.66±0.09
Unrolled GAN, 10 steps [18]	817.4±37.91	1.43±0.12
PacDCGAN2 (ours)	877.1±51.96	0.99±0.13
PacDCGAN3 (ours)	851.6±98.60	1.02±0.34
PacDCGAN4 (ours)	896.0±72.83	0.82±0.25

Table 4.4: Modes covered and KL divergence for unrolled GANs as compared to PacDCGANs for various unrolling parameters, discriminator sizes, and the degree of packing.

CHAPTER 5: THEORETICAL ANALYSES OF PACGAN

In this chapter, we propose a formal and natural mathematical definition of mode collapse, which abstracts away domain-specific details (e.g. images vs. time series). For a target distribution P and a generator distribution Q, this definition describes mode collapse through a two-dimensional representation of the pair (P, Q) as a *region*.

Mode collapse is a phenomenon commonly reported in the GAN literature [32, 15, 39, 47, 38], which can refer to two distinct concepts: (i) the generative model loses some modes that are present in the samples of the target distribution. For example, despite being trained on a dataset of animal pictures that includes lizards, the model never generates images of lizards. (ii) Two distant points in the code vector Z are mapped to the same or similar points in the sample space X. For instance, two distant latent vectors z_1 and z_2 map to the same picture of a lizard [32]. Although these phenomena are different, and either one can occur without the other, they are generally not explicitly distinguished in the literature, and it has been suggested that the latter may cause the former [32]. In this work, we focus on the former notion, as it does not depend on how the generator maps a code vector Z to the sample X, and only focuses on the quality of the samples generated. In other words, we assume here that two generative models with the same marginal distribution over the generated samples should not be treated differently based on how random code vectors are mapped to the data sample space. The second notion of mode collapse would differentiate two such architectures, and is beyond the scope of this work. The proposed region representation relies purely on the properties of the generated samples, and not on the generator's mapping between the latent and sample spaces.

We analyze how the proposed idea of packing changes the training of the generator. We view the discriminator's role as providing a surrogate for a desired loss to be minimized—surrogate in the sense that the actual desired losses, such as Jensen-Shannon divergence or total variation distances, cannot be computed exactly and need to be estimated. Consider the standard GAN discriminator with a cross-entropy loss:

$$\min_{G} \underbrace{\max_{D} \; \mathbb{E}_{X\sim P}[\log(D(X))] - \mathbb{E}_{G(Z)\sim Q}[\log(1 - D(G(Z)))]}_{\simeq \; D_{\mathrm{KL}}\left(P\| \frac{P+Q}{2}\right) + D_{\mathrm{KL}}\left(Q\| \frac{P+Q}{2}\right)} \,, \tag{5.1}$$

where the maximization is over the family of discriminators (or the discriminator weights, if the family is a neural network of a fixed architecture), the minimization is over the family of generators, and X is drawn from the distribution P of the real data, Z is drawn from the distribution of the code vector, typically a low-dimensional Gaussian, and we denote the resulting generator distribution as $G(Z) \sim Q$. The role of the discriminator under this GAN scenario is to provide the generator with an approximation (or a surrogate) of a loss, which in the case of cross entropy loss turns out to be the Jensen-Shannon divergence, defined as $D_{\mathrm{KL}}(P\|(P+Q)/2) + D_{\mathrm{KL}}(Q\|(P+Q)/2)$, where $D_{\mathrm{KL}}(\cdot)$ is the Kullback-Leibler divergence. This follows from the fact that, if we search for the maximizing discriminator over the space of all functions, the maximizer turns out to be $D(X) = P(X)/(P(X)+Q(X))$ [3]. In practice, we search over some parametric family of discriminators, and we can only compute sample average of the losses. This provides an approximation of the Jensen-Shannon divergence between P and Q. The outer minimization over the generator tries to generate samples such that they are close to the real data in this (approximate) Jensen-Shannon divergence, which is one measure of how close the true distribution P and the generator distribution Q are.

In this Chapter, we show a fundamental connection between the principle of packing and mode collapse in GAN. We provide a complete understanding of how packing changes the loss as seen by the generator, by focusing on (as we did to derive the Jensen-Shnnon divergence above) (a) the optimal discriminator over a family of all measurable functions; (b) the population expectation; and (c) the 0-1 loss function of the form:

$$\begin{array}{rl} \max\limits_{D} & \mathbb{E}_{X\sim P}[\mathbb{I}(D(X))] + \mathbb{E}_{G(Z)\sim Q}[1-\mathbb{I}(D(G(Z)))] \\ \text{subject to} & D(X) \in \{0,1\} \ . \end{array}$$

The first assumption allows us to bypass the specific architecture of the discriminator used, which is common when analyzing neural network based discriminators (e.g. [48]). The second assumption can be potentially relaxed and the standard finite sample analysis can be applied to provide bounds similar to those in our main results in Theorems 5.1, 5.2, and 5.3. The last assumption gives a loss of the total variation distance $d_{\mathrm{TV}}(P,Q) \triangleq \sup_{S\subseteq\mathcal{X}}\{P(S)-Q(S)\}$ over the domain $\mathcal{X}$. This follows from the fact that (e.g. [32]),

$$\begin{aligned} & \sup_{D} \big\{\mathbb{E}_{X\sim P}[\mathbb{I}(D(X))] + \mathbb{E}_{G(Z)\sim Q}[1-\mathbb{I}(D(G(Z)))]\big\} \\ = \ & \sup_{S} \big\{P(S) + 1 - Q(S)\big\} \\ = \ & 1 + d_{\mathrm{TV}}(P,Q) \ . \end{aligned}$$

This discriminator provides (an approximation of) the total variation distance, and the generator tries to minimize the total variation distance

$$d_{\mathrm{TV}}(P,Q) \, .$$

The reason we make this assumption is primarily for clarity and analytical tractability: total variation distance highlights the effect of packing in a way that is cleaner and easier to understand than if we were to analyze Jensen-Shannon divergence. We discuss this point in more detail in Chapter 5.2. In sum, these three assumptions allow us to focus purely on the impact of packing on the mode collapse of resulting discriminator.

We want to understand how this 0-1 loss, as provided by such a discriminator, changes with the *degree of packing* m. As packed discriminators see m packed samples, each drawn i.i.d. from one joint class (i.e. either real or generated), we can consider these packed samples as a single sample that is drawn from the product distribution: P^m for real and Q^m for generated. The resulting loss provided by the packed discriminator is therefore $d_{\text{TV}}(P^m, Q^m)$.

We first provide a formal mathematical definition of mode collapse in Chapter 5.1, which leads to a two-dimensional representation of any pair of distributions (P, Q) as a *mode-collapse region*. This region representation provides not only conceptual clarity regarding mode collapse, but also proof techniques that are essential to proving our main results on the fundamental connections between the strength of mode collapse in a pair (P, Q) and the loss $d_{\text{TV}}(P^m, Q^m)$ seen by a packed discriminator (Chapter 5.2). The proofs of these results are provided in Chapter 6. In Chapter 5.2.1, we show that the proposed mode collapse region is equivalent to what is known as the *hypothesis testing region* for type I and type II errors in binary hypothesis testing. This allows us to use strong mathematical techniques from binary hypothesis testing including the data processing inequality and the reverse data processing inequalities.

5.1 MATHEMATICAL DEFINITION OF MODE COLLAPSE AS A TWO-DIMENSIONAL REGION

Although no formal and agreed-upon definition of mode collapse exists in the GAN literature, mode collapse is declared for a multimodal target distribution P if the generator Q assigns a significantly smaller probability density in the regions surrounding a particular subset of modes. One major challenge in addressing such a mode collapse is that it involves the geometry of P: there is no standard partitioning of the domain respecting the modular topology of P, and even heuristic partitions are typically computationally intractable in high dimensions. Hence, we drop this geometric constraint, and introduce a purely analytical definition.

Definition 5.1. *A target distribution P and a generator Q exhibit* (ε, δ)-mode collapse *for some $0 \leq \varepsilon < \delta \leq 1$ if there exists a set $S \subseteq \mathcal{X}$ such that $P(S) \geq \delta$ and $Q(S) \leq \varepsilon$.*

This definition provides a formal measure of mode collapse for a target P and a generator Q; intuitively, larger δ and smaller ε indicate more severe mode collapse. That is, if a large portion of the target $P(S) \geq \delta$ in some set S in the domain $\mathcal{X}$ is missing in the generator $Q(S) \leq \varepsilon$, then we declare (ε, δ)-mode collapse.

A key observation is that *two pairs of distributions can have the same total variation distance while exhibiting very different mode collapse patterns.* To see this, consider a toy example in Figure 5.1, with a uniform target distribution $P = U([0,1])$ over $[0,1]$. Now consider all generators at a fixed total variation distance of 0.2 from P. We compare the intensity of mode collapse for two extreme cases of such generators. $Q_1 = U([0.2,1])$ is uniform over $[0.2,1]$ and $Q_2 = 0.6U([0,0.5]) + 1.4U([0.5,1])$ is a mixture of two uniform distributions, as shown in Figure 5.1. They are designed to have the same total variations distance, i.e. $d_{\mathrm{TV}}(P, Q_1) = d_{\mathrm{TV}}(P, Q_2) = 0.2$, but Q_1 exhibits an extreme mode collapse as the whole probability mass in $[0, 0.2]$ is lost, whereas Q_2 captures a more balanced deviation from P.

Definition 5.1 captures the fact that Q_1 has more mode collapse than Q_2, since the pair (P, Q_1) exhibits $(\varepsilon = 0, \delta = 0.2)$-mode collapse, whereas the pair (P, Q_2) exhibits only $(\varepsilon = 0.12, \delta = 0.2)$-mode collapse, for the same value of $\delta = 0.2$. However, the appropriate way to precisely represent mode collapse (as we define it) is to visualize it through a two-dimensional region we call the *mode collapse region.* For a given pair (P, Q), the corresponding mode collapse region $\mathcal{R}(P, Q)$ is defined as the convex hull of the region of points (ε, δ) such that (P, Q) exhibit (ε, δ)-mode collapse, as shown in Figure 5.1.

$$\mathcal{R}(P,Q) \;\; \triangleq \;\; \mathrm{conv}\Big(\big\{ \, (\varepsilon, \delta) \, \big| \, \delta > \varepsilon \text{ and } (P,Q) \text{ has } (\varepsilon,\delta)\text{-mode collapse} \, \big\} \Big) \, , \qquad (5.2)$$

where $\mathrm{conv}(\cdot)$ denotes the convex hull. This definition of region is fundamental in the sense that it is a sufficient statistic that captures the relations between P and Q. This assertion is made precise in Chapter 5.2.1 by making a strong connection between the mode collapse region and the type I and type II errors in binary hypothesis testing. That connection allows us to prove a sharp result on how the loss, as seen by the discriminator, evolves under PacGAN in Chapter 6. For now, we can use this region representation of a given target-generator pair to detect the strength of mode collapse occurring for a given generator.

Typically, we are interested in the presence of mode collapse with a small ε and a much larger δ; this corresponds to a sharply-increasing slope near the origin $(0,0)$ in the mode collapse region. For example, the middle panel in Figure 5.1 depicts the mode collapse region (shaded in gray) for a pair of distributions (P, Q_1) that exhibit significant mode collapse; notice the sharply-increasing slope at $(0,0)$. The right panel in Figure 5.1 illustrates the same region for a pair of distributions (P, Q_2) that do not exhibit strong mode collapse, resulting a region with a much gentler slope at $(0, 0)$.

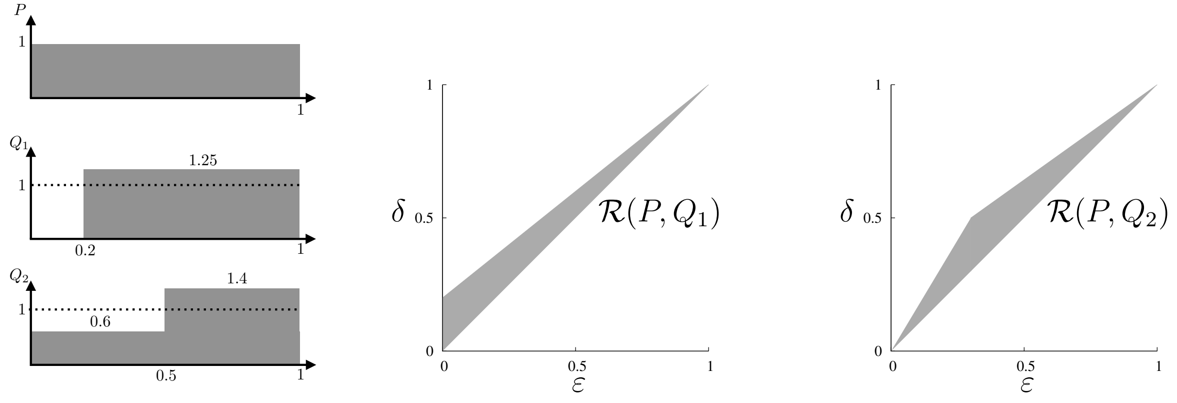


Figure 5.1: A formal definition of (ε, δ)-mode collapse and its accompanying region representation captures the intensity of mode collapse for generators Q_1 with mode collapse and Q_2 which does not have mode collapse, for a toy example distributions P, Q_1, and Q_2 shown on the left. The region of (ε, δ)-mode collapse that is achievable is shown in grey.

Similarly, if the generator assigns a large probability mass compared to the target distribution on a subset, we call it a *mode augmentation*, and give a formal definition below.

Definition 5.2. *A pair of a target distribution P and a generator Q has an (ε, δ)*-mode augmentation *for some $0 \leq \varepsilon < \delta \leq 1$ if there exists a set $S \subseteq \mathcal{X}$ such that $Q(S) \geq \delta$ and $P(S) \leq \varepsilon$.*

Note that we distinguish mode collapse and augmentation strictly here, for analytical purposes. In GAN literature, both collapse and augmentation contribute to the observed "mode collapse" phenomenon, which loosely refers to the lack of diversity in the generated samples.

5.2 EVOLUTION OF THE REGION UNDER PRODUCT DISTRIBUTIONS

The toy example generators Q_1 and Q_2 from Figure 5.1 could not be distinguished using only their total variation distances from P, despite exhibiting very different mode collapse properties. This suggests that the original GAN (with 0-1 loss) may be vulnerable to mode collapse, as it has no way to distinguish distributions in which mode collapse does or does not happen. We prove in Theorem 5.2 that a discriminator that packs multiple samples together *can* distinguish mode-collapsing generators. Intuitively, m packed samples are effectively drawn from the product distributions P^m and Q^m. We show in this section that there is a fundamental connection between the strength of mode collapse of (P, Q) and the loss as seen by the packed discriminator $d_{\mathrm{TV}}(P^m, Q^m)$.

Intuition via toy examples. Concretely, consider the example from the previous section and recall that P^m denote the product distribution resulting from packing together m independent samples from P. Figure 5.2 illustrates how the mode collapse region evolves over m, the degree of packing. This evolution highlights a key insight: the region $\mathcal{R}(P^m, Q_1^m)$ of a mode-collapsing generator expands much faster as m increases compared to the region $\mathcal{R}(P^m, Q_2^m)$ of a non-mode-collapsing generator. This implies that the total variation distance of (P, Q_1) increases more rapidly as we pack more samples, compared to (P, Q_2). This follows from the fact that the total variation distance between P and the generator can be determined directly from the upper boundary of the mode collapse region (see Chapter 5.2.1 for the precise relation). In particular, a larger mode collapse region implies a larger total variation distance between P and the generator. The total variation distances $d_{\mathrm{TV}}(P, Q_1^m)$ and $d_{\mathrm{TV}}(P, Q_2^m)$, which were explicitly chosen to be equal at $m = 1$ in our example, grow farther apart with increasing m, as illustrated in the right figure below. This implies that if we use a packed discriminator, the mode-collapsing generator Q_1 will be heavily penalized for having a larger loss, compared to the non-mode-collapsing Q_2.

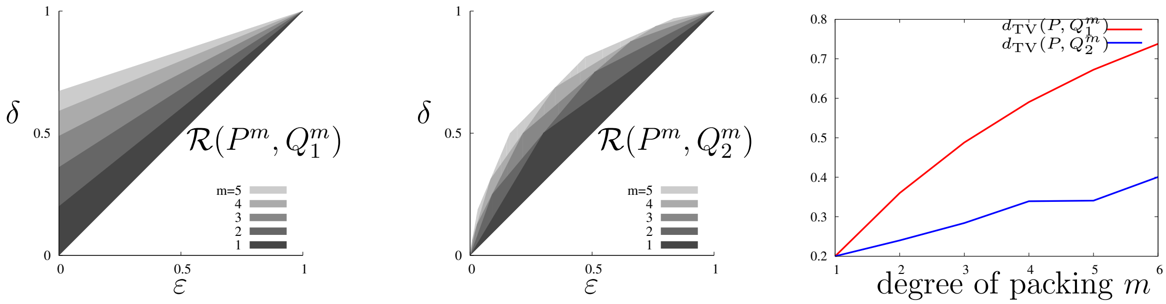


Figure 5.2: Evolution of the mode collapse region over the degree of packing m for the two toy examples from Figure 5.1. The region of the mode-collapsing generator Q_1 expands faster than the non-mode-collapsing generator Q_2 when discriminator inputs are packed (at $m = 1$ these examples have the same TV distances). This causes a discriminator to penalize mode collapse as desired.

Evolution of total variation distances. In order to generalize the intuition from the above toy examples, we first analyze how the total variation evolves for the set of all pairs (P, Q) that have the same total variation distance τ when unpacked (i.e., when $m = 1$). The solutions to the following optimization problems give the desired upper and lower bounds, respectively, on total variation distance for any distribution pair in this set with a packing degree of m:

$$\begin{aligned} \min_{P,Q} &\quad d_{\mathrm{TV}}(P^m, Q^m) & \max_{P,Q} &\quad d_{\mathrm{TV}}(P^m, Q^m) \\ \text{subject to} &\quad d_{\mathrm{TV}}(P, Q) = \tau & \text{subject to} &\quad d_{\mathrm{TV}}(P, Q) = \tau \ , \end{aligned} \tag{5.3}$$

where the maximization and minimization are over all probability measures P and Q. We give the exact solution in Theorem 5.1, which is illustrated pictorially in Figure 5.3 (left).

Theorem 5.1. *For all $0 \leq \tau \leq 1$ and a positive integer m, the solution to the maximization in* (5.3) *is $1 - (1 - \tau)^m$, and the solution to the minimization in* (5.3) *is*

$$L(\tau, m) \triangleq \min_{0 \leq \alpha \leq 1-\tau} d_{\mathrm{TV}}\Big(P_{\mathrm{inner}}(\alpha)^m, Q_{\mathrm{inner}}(\alpha, \tau)^m \Big) \ , \tag{5.4}$$

where $P_{\mathrm{inner}}(\alpha)^m$ and $Q_{\mathrm{inner}}(\alpha, \tau)^m$ are the m-th order product distributions of binary random variables distributed as

$$P_{\mathrm{inner}}(\alpha) = \Big[1 - \alpha, \quad \alpha\Big] \ , \tag{5.5}$$

$$Q_{\mathrm{inner}}(\alpha, \tau) = \Big[1 - \alpha - \tau, \quad \alpha + \tau\Big] \ . \tag{5.6}$$

Although this is a simple statement that can be proved in several different ways, we introduce in Chapter 6 a novel geometric proof technique that critically relies on the proposed mode collapse region. This particular technique will allow us to generalize the proof to more complex problems involving mode collapse in Theorem 5.2, for which other techniques do not generalize. Note that the claim in Theorem 5.1 has nothing to do with mode collapse. Still, we use the mode collapse region definition purely as a proof technique for this claim.

For any given value of τ and m, the bounds in Theorem 5.1 are easy to evaluate numerically, as shown below in the left panel. Within this achievable range, some pairs (P, Q) have rapidly increasing total variation, occupying the upper part of the region (shown in red, middle panel of Figure 5.3), and some pairs (P, Q) have slowly increasing total variation, occupying the lower part as shown in blue in the right panel in Figure 5.3. In particular, the evolution of the mode-collapse region of a pair of m-th power distributions $\mathcal{R}(P^m, Q^m)$ is fundamentally connected to the strength of mode collapse in the original pair (P, Q). This means that for a mode-collapsed pair (P, Q_1), the mth-power distribution will exhibit a different total variation distance evolution than a non-mode-collapsed pair (P, Q_2). As such, these two pairs can be distinguished by a packed discriminator. Making such a claim precise for a broad class of mode-collapsing and non-mode-collapsing generators is challenging, as it depends on the target P and the generator Q, each of which can be a complex high dimensional distribution, like natural images. The proposed region interpretation, endowed with the hypothesis testing interpretation and the data processing inequalities that come with it, is critical: it enables the abstraction of technical details and provides a simple and tight proof based on *geometric techniques* on two-dimensional regions.

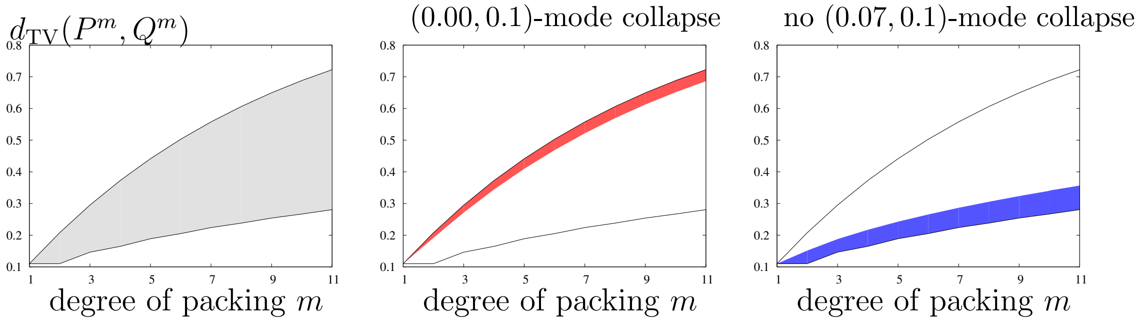


Figure 5.3: The range of $d_{\text{TV}}(P^m, Q^m)$ achievable by pairs with $d_{\text{TV}}(P, Q) = \tau$, for a choice of $\tau = 0.11$, defined by the solutions of the optimization (5.3) provided in Theorem 5.1 (left panel). The range of $d_{\text{TV}}(P^m, Q^m)$ achievable by those pairs that also have $(\varepsilon = 0.00, \delta = 0.1)$-mode collapse (middle panel). A similar range achievable by pairs of distributions that do not have $(\varepsilon = 0.07, \delta = 0.1)$-mode collapse or $(\varepsilon = 0.07, \delta = 0.1)$-mode augmentation (right panel). Pairs (P, Q) with strong mode collapse occupy the top region (near the upper bound) and the pairs with weak mode collapse occupy the bottom region (near the lower bound).

Evolution of total variation distances with mode collapse. We analyze how the total variation evolves for the set of all pairs (P, Q) that have the same total variations distances τ when unpacked, with $m = 1$, and have (ε, δ)-mode collapse for some $0 \leq \varepsilon < \delta \leq 1$. The solution of the following optimization problem gives the desired range of total variation distances:

$$\begin{aligned} \min_{P,Q} \quad & d_{\text{TV}}(P^m, Q^m) & \qquad\qquad \max_{P,Q} \quad & d_{\text{TV}}(P^m, Q^m) \\ \text{subject to} \quad & d_{\text{TV}}(P, Q) = \tau & \text{subject to} \quad & d_{\text{TV}}(P, Q) = \tau \\ (P, Q) \text{ has } & (\varepsilon, \delta)\text{-mode collapse} & (P, Q) \text{ has } & (\varepsilon, \delta)\text{-mode collapse}\ , \end{aligned} \tag{5.7}$$

where the maximization and minimization are over all probability measures P and Q, and the mode collapse constraint is defined in Definition 5.1. (ε, δ)-mode collapsing pairs have total variation at least $\delta - \varepsilon$ by definition, and when $\tau < \delta - \varepsilon$, the feasible set of the above optimization is empty. Otherwise, the next theorem establishes that mode-collapsing pairs occupy the upper part of the total variation region; that is, total variation increases rapidly as we pack more samples together (Figure 5.3, middle panel). This follows from the fact that any pair (P, Q) with total variation distance $\tau \geq \delta - \epsilon$ inherently exhibits (δ, ϵ) mode collapse. One implication is that distribution pairs (P, Q) at the top of the total variation evolution region are those with the strongest mode collapse. Another implication is that a pair (P, Q) with strong mode collapse (i.e., with larger δ and smaller ε in the constraint) will be penalized more under packing, and hence a generator minimizing an approximation of $d_{\text{TV}}(P^m, Q^m)$ will be unlikely to select a distribution that exhibits such strong mode collapse.

Theorem 5.2. *For all $0 \leq \varepsilon < \delta \leq 1$ and a positive integer m, if $1 \geq \tau \geq \delta - \varepsilon$ then the solution to the maximization in* (5.7) *is $1 - (1 - \tau)^m$, and the solution to the minimization in* (5.7) *is*

$$\begin{aligned} L_1(\varepsilon, \delta, \tau, m) \;\; &\triangleq \;\; \min\Bigg\{ \min_{0 \leq \alpha \leq 1 - \frac{\tau\delta}{\delta - \varepsilon}} d_{\text{TV}}\Big(P_{\text{inner1}}(\delta, \alpha)^m, Q_{\text{inner1}}(\varepsilon, \alpha, \tau)^m \Big) \,, \\ &\qquad \min_{1 - \frac{\tau\delta}{\delta - \varepsilon} \leq \alpha \leq 1 - \tau} d_{\text{TV}}\Big(P_{\text{inner2}}(\alpha)^m, Q_{\text{inner2}}(\alpha, \tau)^m \Big) \Bigg\} \,, \end{aligned} \tag{5.8}$$

where $P_{\text{inner1}}(\delta, \alpha)^m$, $Q_{\text{inner1}}(\varepsilon, \alpha, \tau)^m$, $P_{\text{inner2}}(\alpha)^m$, and $Q_{\text{inner2}}(\alpha, \tau)^m$ are the m-th order product distributions of discrete random variables distributed as

$$P_{\text{inner1}}(\delta, \alpha) \;=\; \Big[\delta, \;\; 1 - \alpha - \delta, \;\; \alpha\Big] \,, \tag{5.9}$$

$$Q_{\text{inner1}}(\varepsilon, \alpha, \tau) \;=\; \Big[\varepsilon, \;\; 1 - \alpha - \tau - \varepsilon, \;\; \alpha + \tau\Big] \,, \tag{5.10}$$

$$P_{\text{inner2}}(\alpha) \;=\; \Big[1 - \alpha, \;\; \alpha\Big] \,, \tag{5.11}$$

$$Q_{\text{inner2}}(\alpha, \tau) \;=\; \Big[1 - \alpha - \tau, \;\; \alpha + \tau\Big] \,. \tag{5.12}$$

If $\tau < \delta - \varepsilon$, then the optimization in (5.7) *has no solution and the feasible set is an empty set.*

A proof of this theorem is provided in Chapter 6.1.1, which critically relies on the proposed mode collapse region representation of the pair (P, Q), and the celebrated result by Blackwell from [1]. The solutions in Theorem 5.2 can be numerically evaluated for any given choices of $(\varepsilon, \delta, \tau)$ as we show in Figure 5.4.

Analogous results to the above theorem can be shown for pairs (P, Q) that exhibit (ϵ, δ) mode augmentation (as opposed to mode collapse). These results are omitted for brevity, but the results and analysis are straightforward extensions of the proofs for mode collapse.

This holds because total variation distance is a metric, and therefore symmetric.

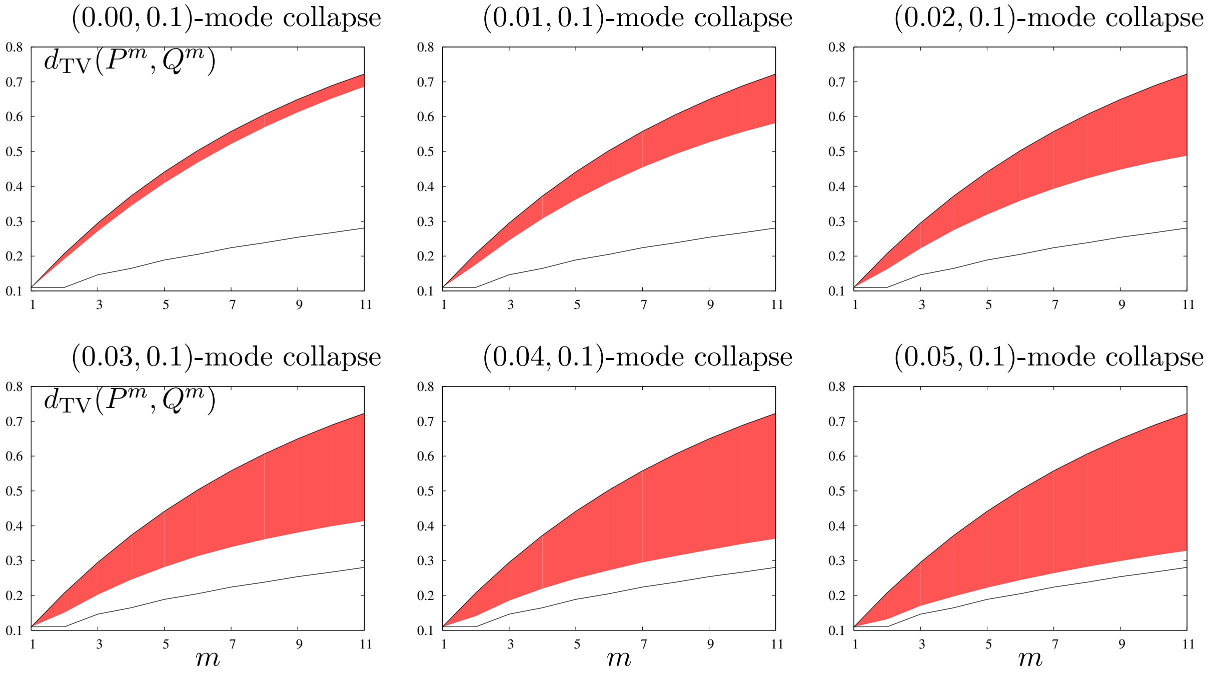


Figure 5.4: The evolution of total variation distance over the packing degree m for mode collapsing pairs is shown as a red band. The upper and lower boundaries of the red band is defined by the optimization 5.7 and computed using Theorem 5.2. For a fixed $d_{\mathrm{TV}}(P, Q) = \tau = 0.11$ and $(\varepsilon, \delta = 0.1)$-mode collapse, we show the evolution with different choices of $\varepsilon \in \{0.00, 0.01, 0.02, 0.03, 0.04, 0.05\}$. The black solid lines show the maximum/minimum total variation in the optimization problem (5.3) as a reference. The family of pairs (P, Q) with stronger mode collapse (i.e. smaller ε in the constraint), occupy a smaller region at the top with higher total variation under packing, and hence is more penalized when training the generator.

Evolution of total variation distances without mode collapse. We next analyze how the total variation evolves for the set of all pairs (P, Q) that have the same total variations distances τ when unpacked, with $m = 1$, and *do not* have (ε, δ)-mode collapse for some $0 \leq \varepsilon < \delta \leq 1$. Because of the symmetry of the total variation distance, mode augmentation in Definition 5.2 is equally damaging as mode collapse, when it comes to how fast total variation distances evolve. Hence, we characterize this evolution for those family of pairs of distributions that do not have either mode collapse or augmentation. The solution of the following optimization problem gives the desired range of total variation distances:

$$\begin{aligned} &\min_{P,Q} \quad d_{\rm TV}(P^m,Q^m) & &\max_{P,Q} \quad d_{\rm TV}(P^m,Q^m) \\ &\text{subject to} \quad d_{\rm TV}(P,Q)=\tau & &\text{subject to} \quad d_{\rm TV}(P,Q)=\tau \\ &(P,Q) \text{ does not have } (\varepsilon,\delta)\text{-mode} & &(P,Q) \text{ does not have } (\varepsilon,\delta)\text{-mode} \\ &\text{collapse or augmentation} & &\text{collapse or augmentation ,} \end{aligned} \tag{5.13}$$

where the maximization and minimization are over all probability measures P and Q, and the mode collapse and augmentation constraints are defined in Definitions 5.1 and 5.2, respectively.

It is not possible to have $d_{\rm TV}(P,Q) > (\delta-\varepsilon)/(\delta+\varepsilon)$ and $\delta+\varepsilon \le 1$ and satisfy the mode collapse and mode augmentation constraints (see Chapter 6.1.2 for a proof). Similarly, it is not possible to have $d_{\rm TV}(P,Q) > (\delta-\varepsilon)/(2-\delta-\varepsilon)$ and $\delta+\varepsilon \ge 1$ and satisfy the constraints. Hence, the feasible set is empty when $\tau > \max\{(\delta-\varepsilon)/(\delta+\varepsilon), (\delta-\varepsilon)/(2-\delta-\varepsilon)\}$. On the other hand, when $\tau \le \delta-\varepsilon$, no pairs with total variation distance τ can have (ε,δ)-mode collapse. In this case, the optimization reduces to the simpler one in (5.3) with no mode collapse constraints. Non-trivial solution exists in the middle regime, i.e. $\delta-\varepsilon \le \tau \le \max\{(\delta-\varepsilon)/(\delta+\varepsilon), (\delta-\varepsilon)/(2-\delta-\varepsilon)\}$. The lower bound for this regime, given in equation (5.17), is the same as the lower bound in (5.4), except it optimizes over a different range of α values. For a wide range of parameters ε, δ, and τ, those lower bounds will be the same, and even if they differ for some parameters, they differ slightly. This implies that the pairs (P,Q) with weak mode collapse will occupy the bottom part of the evolution of the total variation distances (see Figure 5.3 right panel), and also will be penalized less under packing. Hence a generator minimizing (approximate) $d_{\rm TV}(P^m,Q^m)$ is likely to generate distributions with weak mode collapse.

Theorem 5.3. *For all $0 \le \varepsilon < \delta \le 1$ and a positive integer m, if $0 \le \tau < \delta-\varepsilon$, then the maximum and the minimum of* (5.13) *are the same as those of the optimization* (5.3) *provided in Theorem 5.1.*

If $\delta+\varepsilon \le 1$ and $\delta-\varepsilon \le \tau \le (\delta-\varepsilon)/(\delta+\varepsilon)$ then the solution to the maximization in (5.13) *is*

$$\begin{aligned} & U_1(\epsilon,\delta,\tau,m) \\ \triangleq \; & \max_{\alpha+\beta\le 1-\tau, \frac{\varepsilon\tau}{\delta-\varepsilon}\le\alpha,\beta} \; d_{\rm TV}\Big(P_{\rm outer1}(\varepsilon,\delta,\alpha,\beta,\tau)^m, Q_{\rm outer1}(\varepsilon,\delta,\alpha,\beta,\tau)^m\Big) \,, \end{aligned} \tag{5.14}$$

where $P_{\mathrm{outer1}}(\varepsilon,\delta,\alpha,\beta,\tau)^m$ *and* $Q_{\mathrm{outer1}}(\varepsilon,\delta,\alpha,\beta,\tau)^m$ *are the m-th order product distributions of discrete random variables distributed as*

$$P_{\mathrm{outer1}}(\varepsilon,\delta,\alpha,\beta,\tau) \;=\; \left[\tfrac{\alpha(\delta-\varepsilon)-\varepsilon\tau}{\alpha-\varepsilon},\;\; \tfrac{\alpha(\alpha+\tau-\delta)}{\alpha-\varepsilon},\;\; 1-\tau-\alpha-\beta,\;\; \beta,\;\; 0\right],\ \textit{and} \tag{5.15}$$

$$Q_{\mathrm{outer1}}(\varepsilon,\delta,\alpha,\beta,\tau) \;=\; \left[0,\;\; \alpha,\;\; 1-\tau-\alpha-\beta,\;\; \tfrac{\beta(\beta+\tau-\delta)}{\beta-\varepsilon},\;\; \tfrac{\beta(\delta-\varepsilon)-\varepsilon\tau}{\beta-\varepsilon}\right]. \tag{5.16}$$

The solution to the minimization in (5.13) *is*

$$L_2(\tau,m) \;\triangleq\; \min_{\frac{\varepsilon\tau}{\delta-\varepsilon}\le\alpha\le 1-\frac{\delta\tau}{\delta-\varepsilon}} d_{\mathrm{TV}}\Big(P_{\mathrm{inner}}(\alpha)^m, Q_{\mathrm{inner}}(\alpha,\tau)^m\Big), \tag{5.17}$$

where $P_{\mathrm{inner}}(\alpha)$ *and* $Q_{\mathrm{inner}}(\alpha,\tau)$ *are defined as in Theorem 5.1.*

If $\delta+\varepsilon>1$ *and* $\delta-\varepsilon\le\tau\le(\delta-\varepsilon)/(2-\delta-\varepsilon)$ *then the solution to the maximization in* (5.13) *is*

$$\begin{aligned} & U_2(\epsilon,\delta,\tau,m) \\ \triangleq\; & \max_{\alpha+\beta\le 1-\tau,\, \frac{(1-\delta)\tau}{\delta-\varepsilon}\le\alpha,\beta} d_{\mathrm{TV}}\Big(P_{\mathrm{outer2}}(\varepsilon,\delta,\alpha,\beta,\tau)^m, Q_{\mathrm{outer2}}(\varepsilon,\delta,\alpha,\beta,\tau)^m\Big), \end{aligned} \tag{5.18}$$

where $P_{\mathrm{outer1}}(\varepsilon,\delta,\alpha,\beta,\tau)^m$ *and* $Q_{\mathrm{outer1}}(\varepsilon,\delta,\alpha,\beta,\tau)^m$ *are the m-th order product distributions of discrete random variables distributed as*

$$P_{\mathrm{outer2}}(\varepsilon,\delta,\alpha,\beta,\tau) \;=\; \left[\tfrac{\alpha(\delta-\varepsilon)-(1-\delta)\tau}{\alpha-(1-\delta)},\;\; \tfrac{\alpha(\alpha+\tau-(1-\varepsilon))}{\alpha-(1-\delta)},\;\; 1-\tau-\alpha-\beta,\;\; \beta,\;\; 0\right]\ ,\ \textit{and} \tag{5.19}$$

$$Q_{\mathrm{outer2}}(\varepsilon,\delta,\alpha,\beta,\tau) \;=\; \left[0,\;\; \alpha,\;\; 1-\tau-\alpha-\beta,\;\; \tfrac{\beta(\beta+\tau-(1-\varepsilon))}{\beta-(1-\delta)},\;\; \tfrac{\beta(\delta-\varepsilon)-(1-\delta)\tau}{\beta-(1-\delta)}\right]. \tag{5.20}$$

The solution to the minimization in (5.13) *is*

$$L_3(\tau,m) \;\triangleq\; \min_{\frac{(1-\delta)\tau}{\delta-\varepsilon}\le\alpha\le 1-\frac{(1-\varepsilon)\tau}{\delta-\varepsilon}} d_{\mathrm{TV}}\Big(P_{\mathrm{inner}}(\alpha)^m, Q_{\mathrm{inner}}(\alpha,\tau)^m\Big), \tag{5.21}$$

where $P_{\mathrm{inner}}(\alpha)$ *and* $Q_{\mathrm{inner}}(\alpha,\tau)$ *are defined as in Theorem 5.1.*

If $\tau>\max\{(\delta-\varepsilon)/(\delta+\varepsilon),(\delta-\varepsilon)/(2-\delta-\varepsilon)\}$, *then the optimization in* (5.13) *has no solution and the feasible set is an empty set.*

A proof of this theorem is provided in Chapter 6.1.2, which also critically relies on the proposed mode collapse region representation of the pair (P, Q) and the celebrated result by Blackwell from [1]. The solutions in Theorem 5.3 can be numerically evaluated for any given choices of $(\varepsilon, \delta, \tau)$ as we show in Figure 5.5.

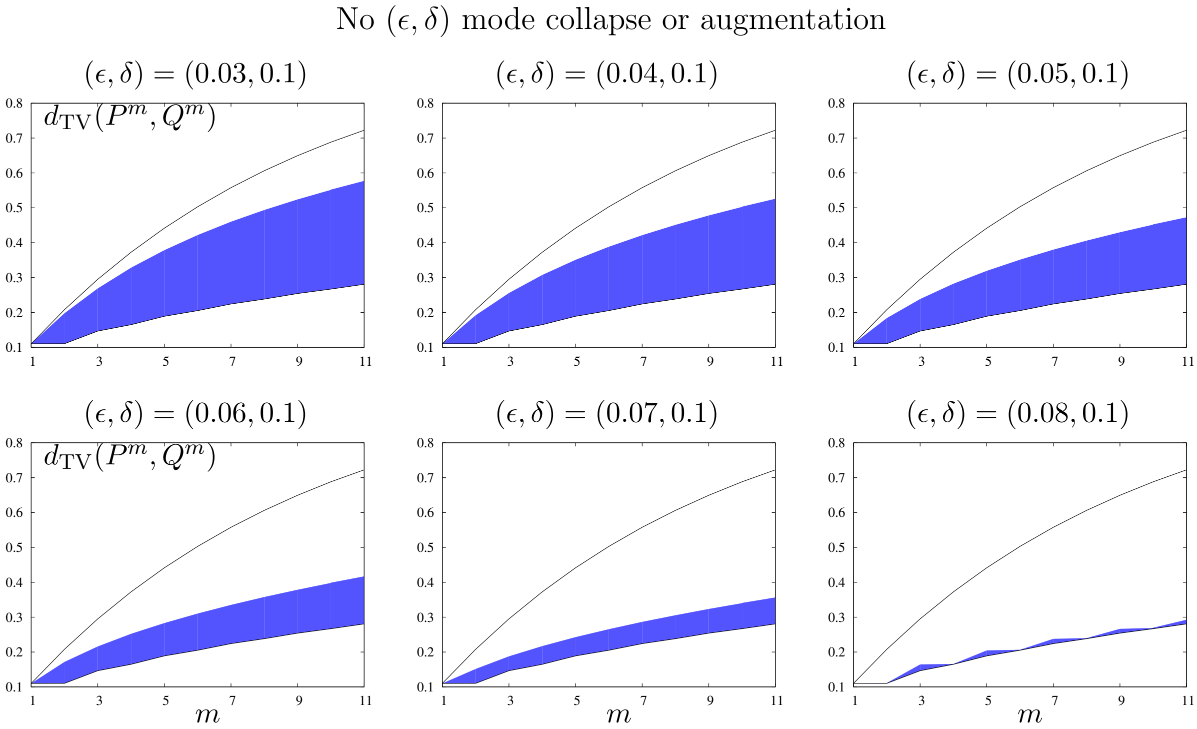


Figure 5.5: The evolution of total variation distance over the packing degree m for pairs with no mode collapse is shown as a blue band, as defined by the optimization (5.13) and computed using Theorem 5.3. For a fixed $d_{\mathrm{TV}}(P, Q) = \tau = 0.11$ and the lack of $(\varepsilon, \delta = 0.1)$-mode collapse constraints, we show the evolution with different choices of $\varepsilon \in \{0.03, 0.04, 0.05, 0.06, 0.07, 0.08\}$. The black solid lines show the maximum/minimum total variation in the optimization (5.3) as a reference. The family of pairs (P, Q) with weaker mode collapse (i.e. larger ε in the constraint), occupies a smaller region at the bottom with smaller total variation under packing, and hence is less penalized when training the generator.

The benefit of packing degree m. We give a practitioner the choice of the degree m of packing, namely how many samples to jointly pack together. There is a natural trade-off between computational complexity (which increases gracefully with m) and the additional distinguishability, which we illustrate via an example. Consider the goal of differentiating two families of target-generator pairs, one with mode collapse and one without:

$$\begin{aligned}
&H_0(\varepsilon,\delta,\tau) \triangleq \\
&\{(P,Q)|(P,Q) \text{ without } (\varepsilon,\delta)\text{-mode collapse or augmentation,} \\
&\text{ and } d_{\mathrm{TV}}(P,Q)=\tau\} \ , \\
&H_1(\varepsilon,\delta,\tau) \triangleq \{(P,Q)|(P,Q) \text{ with } (\varepsilon,\delta)\text{-mode collapse and } d_{\mathrm{TV}}(P,Q)=\tau\} \ . \qquad (5.22)
\end{aligned}$$

As both families have the same total variation distances, they cannot be distinguished by an unpacked discriminator. However, a packed discriminator that uses m samples jointly can differentiate those two classes and even separate them entirely for a certain choices of parameters, as illustrated in Figure 5.6. In red, we show the achievable $d_{\mathrm{TV}}(P^m, Q^m)$ for $H_1(\varepsilon = 0.02, \delta = 0.1, \tau = 0.11)$ (the bounds in Theorem (5.2)). In blue is shown a similar region for $H_0(\varepsilon = 0.05, \delta = 0.1, \tau = 0.11)$ (the bounds in Theorem (5.3)). Although the two families are strictly separated (one with $\varepsilon = 0.02$ and another with $\varepsilon = 0.05$), a non-packed discriminator cannot differentiate those two families as the total variation is the same for both. However, as you pack mode samples, the packed discriminator becomes more powerful in differentiating the two hypothesized families. For instance, for $m \geq 5$, the total variation distance completely separates the two families.

In general, the overlap between those regions depends on the specific choice of parameters, but the overall trend is universal: packing separates generators with mode collapse from those without. Further, as the degree of packing increases, a packed discriminator increasingly penalizes generators with mode collapse and rewards generators that exhibit less mode collapse. Even if we consider complementary sets H_0 and H_1 with the same ε and δ (such that the union covers the whole space of pairs of (P, Q) with the same total variation distance), the least penalized pairs will be those with least mode collapse, which fall within the blue region of the bottom right panel in Figure 5.5. This is consistent with the empirical observations in Tables 4.1, 4.3, and 4.4, where increasing the degree of packing captures more modes.

Jensen-Shannon divergence. In this theoretical analysis, we have focused on 0-1 loss, as our current analysis technique gives exact solutions to the optimization problems (5.3), (5.7), and (5.13) if the metric is total variation distance. This follows from the fact that we can provide tight inner and outer regions to the family of mode collapse regions $\mathcal{R}(P, Q)$ that have the same total variation distances as $d_{\mathrm{TV}}(P, Q)$ as shown in Chapter 6.

In practice, 0-1 loss is never used, as it is not differentiable. The most popular choice of a loss function is cross entropy loss, which gives a metric of Jensen-Shannon (JS) divergence,

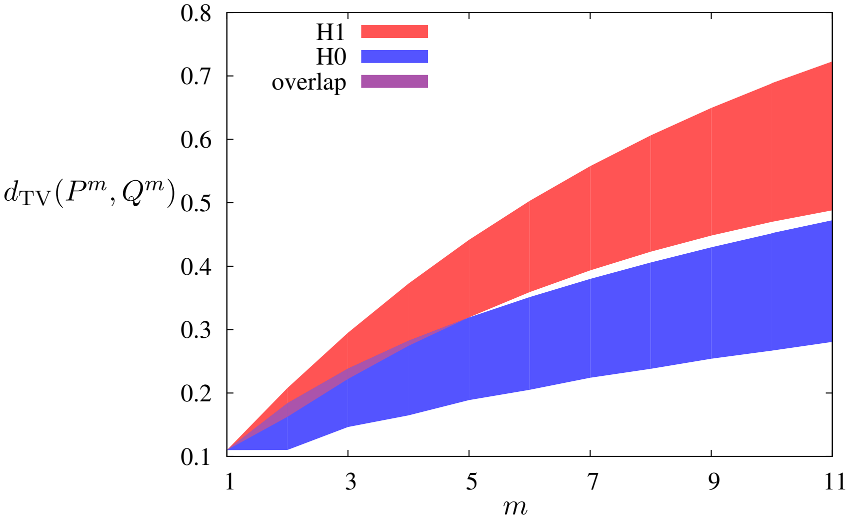


Figure 5.6: Evolution of achievable total variation distances $d_{\mathrm{TV}}(P^m, Q^m)$ over packing size m for two families of the target-generator pairs $H_0(0.05, 0.1, 0.11)$ and $H_1(0.02, 0.1, 0.11)$. The mode-collapsing H_1 is penalized significantly by the discriminator (only with $m > 1$) and the two families can be strictly separated with packing for $m > 5$.

as shown in the beginning of Chapter 5. However, the same proof techniques used to show Theorems 5.2 and 5.3 give loose bounds on JS divergence. In particular, this gap prevents us from sharply characterizing the full effect of packing degree m on the JS divergence of a pair of distributions. Nonetheless, we find that empirically, packing seems to reduce mode collapse even under a cross entropy loss. Hence, we leave it as a future research direction to find solutions to the optimization problems (5.3), (5.7), and (5.13), when the metric is the (more common) Jensen-Shannon divergence.

5.2.1 Operational interpretation of mode collapse via hypothesis testing region

So far, all the definitions and theoretical results have been explained without explicitly using the *mode collapse region*. The main contribution of introducing the region definition is that it provides a new proof technique based on the geometric properties of these two-dimensional regions. Concretely, we show that the proposed mode collapse region is equivalent to a similar notion in binary hypothesis testing. This allows us to bring powerful mathematical tools from this mature area in statistics and information theory—in particular, the *data processing inequalities* originating from the seminal work of Blackwell [1]. We make this connection precise, which gives insights on how to interpret the mode collapse region, and list the properties and techniques which dramatically simplify the proof, while providing the tight results in Chapter 6.

5.2.2 Equivalence between the mode collapse region and the hypothesis testing region

There is a simple one-to-one correspondence between mode collapse region as we define it in Chapter 5.1 (e.g. Figure 5.1) and the hypothesis testing region studied in binary hypothesis testing. In the classical testing context, there are two hypotheses, $h = 0$ or $h = 1$, and we make observations via some stochastic experiment in which our observations depend on the hypothesis. Let X denote this observation. One way to visualize such an experiment is using a two-dimensional region defined by the corresponding type I and type II errors. This was, for example, used to prove strong composition theorems in the context of differential privacy in [31], and subsequently to identify the optimal differentially private mechanisms under local privacy [29] and multi-party communications [30]. We refer to [31] for the precise definition of the hypothesis testing region and its properties.

We can map this binary hypothesis testing setup directly to the GAN context. Suppose the null hypothesis $h = 0$ denotes observations being drawn from the true distribution P, and the alternate hypothesis $h = 1$ denotes observations being drawn from the generated distribution Q. Given a sample X from this experiment, suppose we make a decision on whether the sample came from P or Q based on a rejection region S_{reject}, such that we reject the null hypothesis if $X \in S_{\text{reject}}$. Type I error is when the null hypothesis is true but rejected, which happens with $\mathbb{P}(X \in S_{\text{reject}}|h = 0)$, and type II error is when the null hypothesis is false but accepted, which happens with $\mathbb{P}(X \notin S_{\text{reject}}|h = 1)$. Sweeping through the achievable pairs $(\mathbb{P}(X \notin S_{\text{reject}}|h = 1), \mathbb{P}(X \in S_{\text{reject}}|h = 0))$ for all possible rejection sets, this defines a two dimensional convex region that is called *hypothesis testing region*. An example of hypothesis testing regions for the two toy examples (P, Q_1) and (P, Q_2) from Figure 5.1 are shown below in Figure 5.7.

In defining the region, we allow stochastic decisions, such that if a point (x, y) and another point (x', y') are achievable type II and type I errors, then any convex combination of those points are also achievable by randomly choosing between those two rejection sets. Hence, the resulting hypothesis testing region is always a convex set by definition. We also show only the region below the 45-degree line passing through $(1, 0)$ and $(1, 0)$, as the other region is symmetric and redundant. For a given pair (P, Q), there is a very simple relation between its mode collapse region and hypothesis testing region.

Remark 5.1 (Equivalence)**.** *For a pair of target P and generator Q, the hypothesis testing region is a mirror image of the mode collapse region with respect to a horizontal axis at*

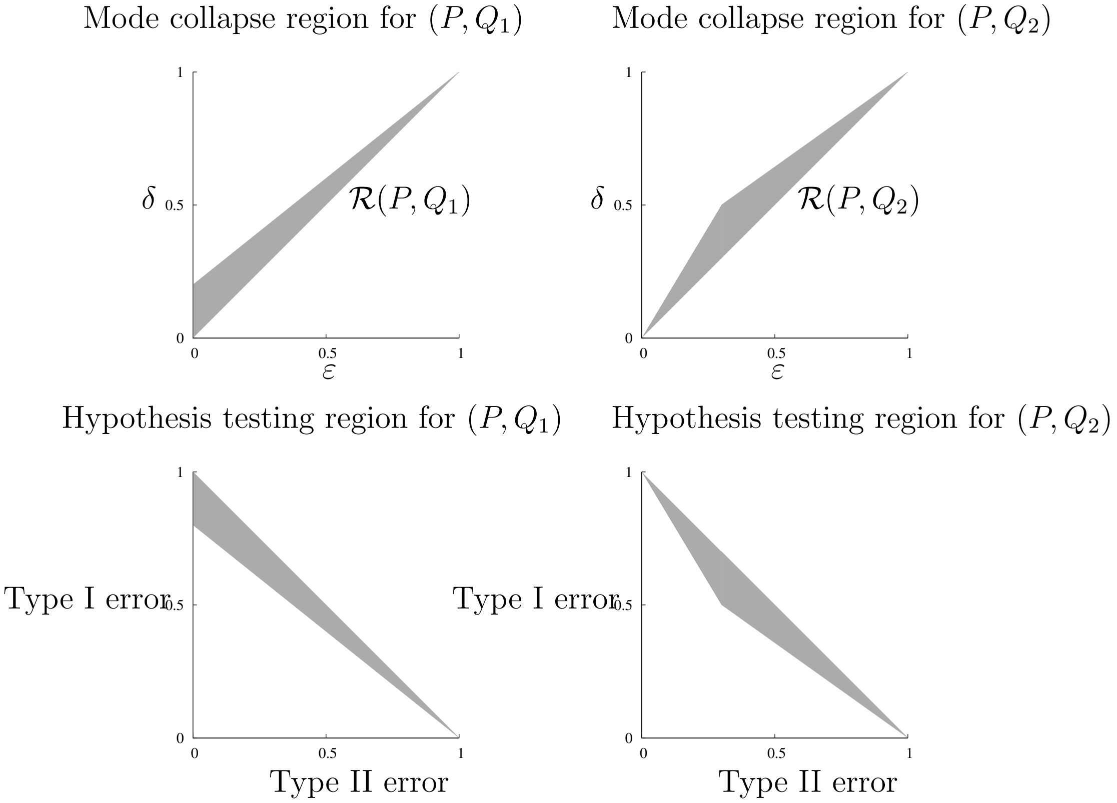


Figure 5.7: The hypothesis testing region of (P, Q) (bottom row) is a mirror image of the mode collapse region (top row). We omit the region above $y = 1 - x$ axis in the hypothesis testing region as it is symmetric. The regions for mode collapsing toy example in Figure 5.1 (P, Q_1) are shown on the left and the regions for the non mode collapsing example (P, Q_2) are shown on the right.

$\delta = 1/2$.

For example, the hypothesis testing regions of the toy examples from Figure 5.1 are shown below in Figure 5.7. This simple relation allows us to tap into the rich analysis tools known for hypothesis testing regions. We list such properties of mode collapse regions derived from this relation in the next Chapter. The proof of all the remarks follow from the equivalence to binary hypothesis testing and corresponding existing results from [1] and [31].

5.2.3 Properties of the mode collapse region

Given the equivalence between the mode collapse region and the binary hypothesis testing region, several important properties follow as corollaries. First, the mode collapse region $\mathcal{R}(P, Q)$ is a convex set, by definition. Second, the hypothesis testing region is a sufficient statistic for the purpose of binary hypothesis testing from a pair of distributions (P, Q). This implies, among other things, that all f-divergences can be computed from the region. In particular, for the purpose of GAN with 0-1 loss, we can define total variation as a geometric property of the region, which is crucial to proving our main results.

Remark 5.2 (Total variation distance)**.** *The total variation distance between P and Q is the intersection between the vertical axis and the tangent line to the upper boundary of $\mathcal{R}(P, Q)$ that has a slope of one, as shown in Figure 5.8.*

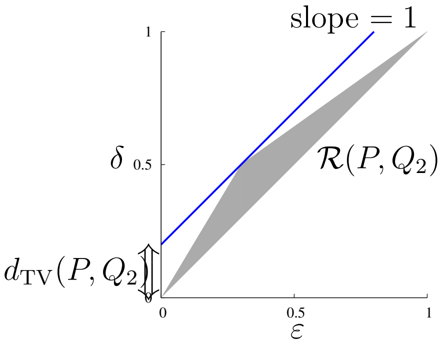


Figure 5.8: Total variation distance is one among many properties of (P, Q_2) that can be directly read off of the region $\mathcal{R}(P, Q)$.

This follows from the equivalence of the mode collapse region (Remark 5.1) and the hypothesis testing region. This geometric definition of total variation allows us to enumerate over all pairs (P, Q) that have the same total variation τ in our proof, via enumerating over all regions that touch the line that has a unit slope and a shift τ (see Figure 6.1).

The major strength of the region perspective, as originally studied by Blackwell [1], is in providing a comparison of stochastic experiments. In our GAN context, consider comparing two pairs of target distributions and generators (P, Q) and (P', Q') as follows. First, a hypothesis h is drawn, choosing whether to produce samples from the true distribution, in which case we say $h = 0$, or to produce samples from the generator, in which case we say $h = 1$. Conditioned on this hypothesis h, we use X to denote a random variable that is drawn from the first pair (P, Q) such that $f_{X|h}(x|0) = P(x)$ and $f_{X|h}(x|1) = Q(x)$. Similarly, we use X' to denote a random sample from the second pair, where $f_{X'|h}(x|0) = P'(x)$ and $f_{X'|h}(x|1) = Q'(x)$. Note that the conditional distributions are well-defined for both X and X', but there is no coupling defined between them. We can without loss of generality assume h to be independently drawn from the uniform distribution.

Definition 5.3. *For a given coupling between X and X', we say X* dominates *X' if they form a Markov chain h–X–X'.*

The *data processing inequality* in the following remark shows that if we further *process* the output samples from the pair (P, Q) then the further processed samples can only have less mode collapse. Processing output of stochastic experiments has the effect of smoothing out the distributions, and mode collapse, which corresponds to a *peak* in the pair of distributions, are smoothed out in the processing down the Markov chain.

Remark 5.3 (Data processing inequality)**.** *The following data processing inequality holds for the mode collapse region. For two coupled target-generator pairs (P, Q) and (P', Q'), if X* dominates *another pair X', then*

$$\mathcal{R}(P', Q') \;\; \subseteq \;\; \mathcal{R}(P, Q) \; .$$

This is expected, and follows directly from the equivalence of the mode collapse region (Remark 5.1) and the hypothesis testing region. What is perhaps surprising is that the reverse is also true.

Remark 5.4 (Reverse data processing inequality)**.** *The following reverse data processing inequality holds for the mode collapse region. For two paired marginal distributions X and X', if*

$$\mathcal{R}(P', Q') \;\; \subseteq \;\; \mathcal{R}(P, Q) \; ,$$

then there exists a coupling of the random samples from X and X' such that X dominates X', i.e. they form a Markov chain h–X–X'.

This follows from the equivalence between the mode collapse region and the hypothesis testing region (Remark 5.1) and Blackwell's celebrated result on comparisons of stochastic experiments [1]. This region interpretation, and the accompanying (reverse) data processing inequality, abstracts away all the details about P and Q, enabling us to use geometric analysis tools to prove our results. In proving our main results, we will mainly rely on the following remark, which is the corollary of the Remarks 5.3 and 5.4.

Remark 5.5. *For all positive integers m, the dominance of regions are preserved under taking m-th order product distributions, i.e. if $\mathcal{R}(P', Q') \subseteq \mathcal{R}(P, Q)$, then $\mathcal{R}((P')^m, (Q')^m) \subseteq \mathcal{R}(P^m, Q^m)$.*

CHAPTER 6: PROOFS OF THE MAIN RESULTS

In this Chapter, we showcase how the region interpretation provides a new proof technique that is simple and tight. This transforms the measure-theoretic problem into a geometric one in a simple 2D compact plane, facilitating the proof of otherwise-challenging results.

6.1 PROOF OF THE MAIN THEOREM

Note that although the original optimization (5.3) has nothing to do with mode collapse, we use the mode collapse region to represent the pairs (P, Q) to be optimized over. This allows us to use simple geometric techniques to enumerate over all possible pairs (P, Q) that have the same total variation distance τ.

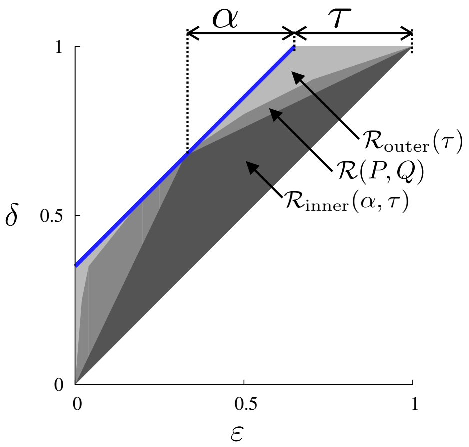


Figure 6.1: For any pair (P, Q) with total variation distance τ, there exists an α such that the corresponding region $\mathcal{R}(P, Q)$ is sandwiched between $\mathcal{R}_{\text{inner}}(\alpha, \tau)$ and $\mathcal{R}_{\text{outer}}(\tau)$.

By Remark 5.2, all pairs (P, Q) that have total variation τ must have a mode collapse region $\mathcal{R}(P, Q)$ that is tangent to the blue line in Figure 6.1. Let us denote a point where $\mathcal{R}(P, Q)$ meets the blue line by the point $(1 - \alpha - \tau, 1 - \alpha)$ in the 2D plane, parametrized by $\alpha \in [0, 1 - \tau]$. Then, for any such (P, Q), we can sandwich the region $\mathcal{R}(P, Q)$ between two regions $\mathcal{R}_{\text{inner}}$ and $\mathcal{R}_{\text{outer}}$:

$$\mathcal{R}_{\text{inner}}(\alpha, \tau) \;\subseteq\; \mathcal{R}(P, Q) \;\subseteq\; \mathcal{R}_{\text{outer}}(\tau) \;, \tag{6.1}$$

which are illustrated in Figure 6.2. Now, we wish to understand how these inner and outer regions evolve under product distributions. This endeavor is complicated by the fact that there can be infinite pairs of distributions that have the same region $\mathcal{R}(P,Q)$. However, note that if two pairs of distributions have the same region $\mathcal{R}(P,Q)=\mathcal{R}(P',Q')$, then their product distributions will also have the same region $\mathcal{R}(P^m,Q^m)=\mathcal{R}((P')^m,(Q')^m)$. As such, we can focus on the simplest, *canonical* pair of distributions, whose support set has the minimum cardinality over all pairs of distributions with region $\mathcal{R}(P,Q)$.

For a given α, we denote the pairs of canonical distributions achieving these exact inner and outer regions as in Figure 6.2: let $(P_{\text{inner}}(\alpha), Q_{\text{inner}}(\alpha,\tau))$ be as defined in (5.5) and (5.6), and let $(P_{\text{outer}}(\tau), Q_{\text{outer}}(\tau))$ be defined as below. Since the outer region has three sides (except for the universal 45-degree line), we only need alphabet size of three to find the canonical probability distributions corresponding to the outer region. By the same reasoning, the inner region requires only a binary alphabet. Precise probability mass functions on these discrete alphabets can be found easily from the shape of the regions and the equivalence to the hypothesis testing region explained in Chapter 5.2.1.

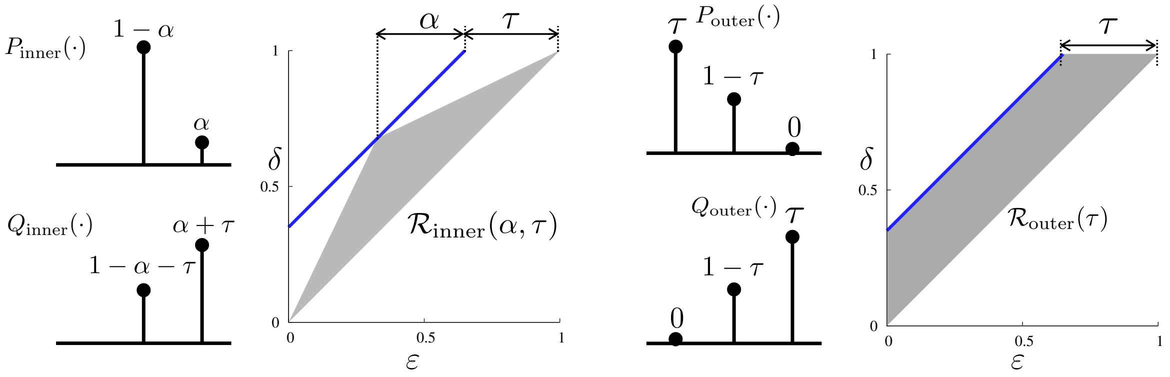


Figure 6.2: Canonical pairs of distributions corresponding to $\mathcal{R}_{\text{inner}}(\alpha,\tau)$ and $\mathcal{R}_{\text{outer}}(\tau)$.

By the preservation of dominance under product distributions in Remark 5.5, it follows from the dominance in (6.1) that for any (P,Q) there exists an α such that

$$\begin{aligned}\mathcal{R}(P_{\text{inner}}(\alpha)^m, Q_{\text{inner}}(\alpha,\tau)^m) \;\; \subseteq \;\; \mathcal{R}(P^m,Q^m) \;\; \subseteq \\ \mathcal{R}(P_{\text{outer}}(\tau)^m, Q_{\text{outer}}(\tau)^m)\;.\end{aligned} \tag{6.2}$$

Due to the data processing inequality of mode collapse region in Remark 5.4, it follows that dominance of region implies dominance of total variation distances:

$$\begin{aligned}&\min\nolimits_{0\le\alpha\le1-\tau} d_{\text{TV}}(P_{\text{inner}}(\alpha)^m, Q_{\text{inner}}(\alpha,\tau)^m) \\ &\qquad \le \;\; d_{\text{TV}}(P^m,Q^m) \;\; \le \qquad\qquad d_{\text{TV}}(P_{\text{outer}}(\tau)^m, Q_{\text{outer}}(\tau)^m)\;.\end{aligned} \tag{6.3}$$

The RHS and LHS of the above inequalities can be completely characterized by taking the m-th power of those canonical pairs of distributions. For the upper bound, all mass except for $(1-\tau)^m$ is nonzero only on one of the pairs, which gives $d_{\rm TV}(P^m_{\rm outer}, Q^m_{\rm outer}) = 1-(1-\tau)^m$. For the lower bound, writing out the total variation gives $L(\tau, m)$ in (5.4). This finishes the proof of Theorem 5.1.

6.1.1 Proof of Theorem 5.2

In optimization (5.7), we consider only those pairs with (ε, δ)-mode collapse. It is simple to see that the outer bound does not change. We only need a new inner bound. Let us denote a point where $\mathcal{R}(P, Q)$ meets the blue line by the point $(1-\alpha-\tau, 1-\alpha)$ in the 2D plane, parametrized by $\alpha \in [0, 1-\tau]$. We consider the case where $\alpha < 1-(\tau\delta/(\delta-\varepsilon))$ for now, and treat the case when α is larger separately, as the analyses are similar but require a different canonical pair of distributions (P, Q) for the inner bound. The additional constraint that (P, Q) has (ε, δ)-mode collapse translates into a geometric constraint that we need to consider all regions $\mathcal{R}(P, Q)$ that include the orange solid circle at point (ε, δ). Then, for any such (P, Q), we can sandwich the region $\mathcal{R}(P, Q)$ between two regions $\mathcal{R}_{\rm inner1}$ and $\mathcal{R}_{\rm outer}$:

$$\mathcal{R}_{\rm inner1}(\varepsilon, \delta, \alpha, \tau) \;\subseteq\; \mathcal{R}(P, Q) \;\subseteq\; \mathcal{R}_{\rm outer}(\tau) \;, \tag{6.4}$$

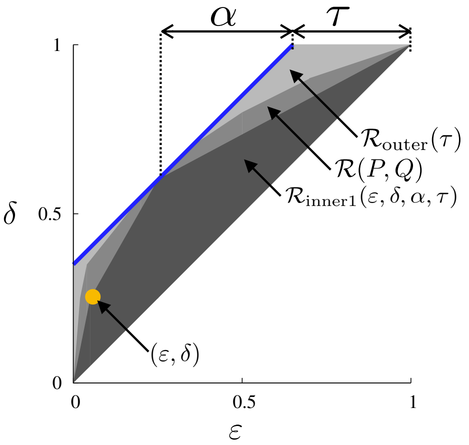

α
τ
1
δ
0.5
0
0
0.5
1
ε
$\mathcal{R}_{\rm outer}(\tau)$
$\mathcal{R}(P, Q)$
$\mathcal{R}_{\rm inner1}(\varepsilon, \delta, \alpha, \tau)$
(ε, δ)

Figure 6.3: For any pair (P, Q) with (ε, δ)-mode collapse, the corresponding region $\mathcal{R}(P, Q)$ is sandwiched between $\mathcal{R}_{\rm inner1}(\varepsilon, \delta, \alpha, \tau)$ and $\mathcal{R}_{\rm outer}(\tau)$.

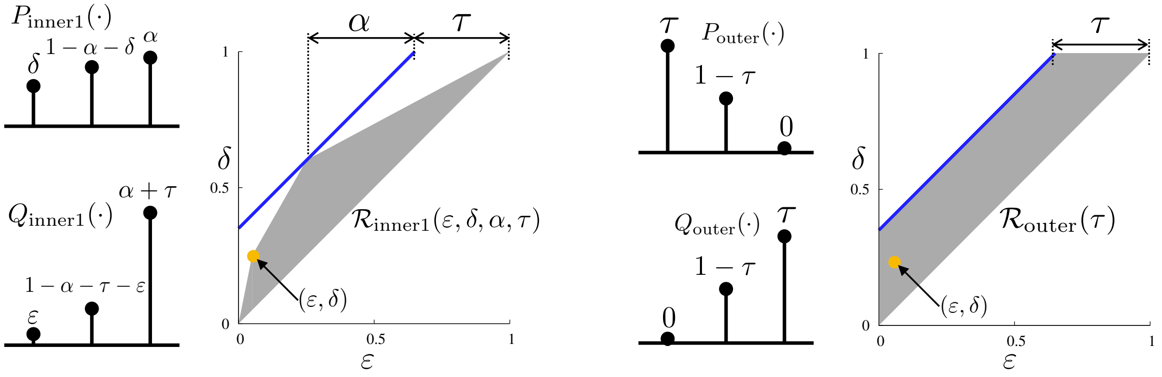
Let

$$(P_{\rm inner1}(\delta, \alpha), Q_{\rm inner1}(\varepsilon, \alpha, \tau))$$

defined in (5.9) and (5.10), and

$$(P_{\text{outer}}(\tau), Q_{\text{outer}}(\tau))$$

defined in Chapter 6.1 denote the pairs of canonical distributions achieving the inner and outer regions exactly as shown in Figure 6.4. By the preservation of dominance under product distributions in Remark 5.5, it follows from the dominance in (6.4) that for any (P,Q) there exists an α such that



$P_{\text{inner1}}(\cdot)$
δ
$1-\alpha-\delta$
α
$Q_{\text{inner1}}(\cdot)$
$\alpha+\tau$
$1-\alpha-\tau-\varepsilon$
ε
α
τ
1
0.5
0
δ
$\mathcal{R}_{\text{inner1}}(\varepsilon,\delta,\alpha,\tau)$
(ε,δ)
0
0.5
1
ε
τ
$P_{\text{outer}}(\cdot)$
$1-\tau$
0
$Q_{\text{outer}}(\cdot)$
τ
$1-\tau$
0
τ
1
0.5
0
δ
$\mathcal{R}_{\text{outer}}(\tau)$
(ε,δ)
0
0.5
1
ε

Figure 6.4: Canonical pairs of distributions corresponding to $\mathcal{R}_{\text{inner}}(\varepsilon,\delta,\tau,\alpha)$ and $\mathcal{R}_{\text{outer}}(\tau)$.

$$\begin{aligned} \mathcal{R}(P_{\text{inner1}}(\delta,\alpha)^m, Q_{\text{inner1}}(\varepsilon,\delta,\alpha,\tau)^m) \;&\subseteq\; \mathcal{R}(P^m,Q^m) \\ &\subseteq\; \mathcal{R}(P_{\text{outer}}(\tau)^m, Q_{\text{outer}}(\tau)^m)\;. \end{aligned} \tag{6.5}$$

Due to the data processing inequality of mode collapse region in Remark 5.4, it follows that dominance of region implies dominance of total variation distances:

$$\begin{aligned} \min_{0\leq\alpha\leq 1-\frac{\tau\delta}{\delta-\varepsilon}} d_{\text{TV}}(P_{\text{inner1}}(\delta,\alpha)^m, Q_{\text{inner1}}(\varepsilon,\delta,\alpha,\tau)^m) \leq \;\; d_{\text{TV}}(P^m,Q^m) \;\;\leq \\ d_{\text{TV}}(P_{\text{outer}}(\tau)^m, Q_{\text{outer}}(\tau)^m)\;. \end{aligned} \tag{6.6}$$

The RHS and LHS of the above inequalities can be completely characterized by taking the m-th power of those canonical pairs of distributions. For the upper bound, all mass except for $(1-\tau)^m$ is nonzero only on one of the pairs, which gives $d_{\text{TV}}(P^m_{\text{outer}}, Q^m_{\text{outer}}) = 1-(1-\tau)^m$. For the lower bound, writing out the total variation gives $L_1(\varepsilon,\delta,\tau,m)$ in (5.8).

For $\alpha > 1-(\tau\delta/(\delta-\varepsilon))$, we need to consider a different class of canonical distributions for the inner region, shown below. The inner region $\mathcal{R}_{\text{inner2}}(\alpha,\tau)$ and corresponding canonical distributions $P_{\text{inner2}}(\alpha)$ and $Q_{\text{inner2}}(\alpha,\tau)$ defined in (5.11) and (5.12) are shown below. We take the smaller one between the total variation distance resulting from these two cases. Note that $\alpha \leq 1-\tau$ by definition. This finishes the proof of Theorem 5.2.

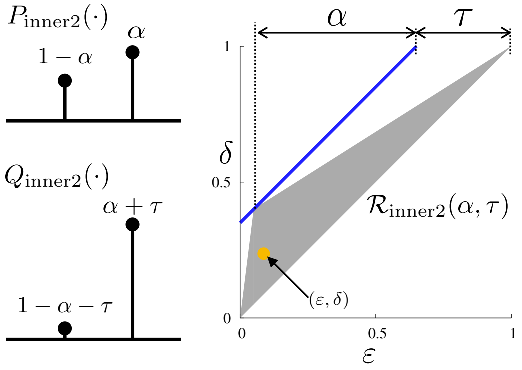


Figure 6.5: When $\alpha > 1-(\tau\delta/(\delta-\varepsilon))$, this shows a canonical pair of distributions corresponding to $\mathcal{R}_{\text{inner}}(\varepsilon,\delta,\tau,\alpha)$ for the mode-collapsing scenario $H_1(\varepsilon,\delta,\tau)$.

6.1.2 Proof of Theorem 5.3

When $\tau < \delta-\varepsilon$, all pairs (P,Q) with $d_{\text{TV}}(P,Q)=\tau$ cannot have (ε,δ)-mode collapse, and the optimization of (5.13) reduces to that of (5.3) without any mode collapse constraints.

When $\delta+\varepsilon \leq 1$ and $\tau > (\delta-\varepsilon)/(\delta+\varepsilon)$, no convex region $\mathcal{R}(P,Q)$ can touch the 45-degree line at τ as shown below, and the feasible set is empty. This follows from the fact that a triangle region passing through both (ε,δ) and $(1-\delta,1-\varepsilon)$ will have a total variation distance of $(\delta-\varepsilon)/(\delta+\varepsilon)$. Note that no (ε,δ) mode augmentation constraint translates into the region not including the point $(1-\delta,1-\varepsilon)$. We can see easily from Figure 6.6 that any total variation beyond that will require violating either the no-mode-collapse constraint or the no-mode-augmentation constraint. Similarly, when $\delta+\varepsilon > 1$ and $\tau > (\delta-\varepsilon)/(2-\delta-\varepsilon)$, the feasible set is also empty. These two can be unified as $\tau > \max\{(\delta-\varepsilon)/(\delta+\varepsilon),(\delta-\varepsilon)/(2-\delta-\varepsilon)\}$.

Suppose $\delta+\varepsilon \leq 1$, and consider the intermediate regime when $\delta-\varepsilon \leq \tau \leq (\delta-\varepsilon)/(\delta+\varepsilon)$. In optimization (5.13), we consider only those pairs with no (ε,δ)-mode collapse or (ε,δ)-mode augmentation. It is simple to see that the inner bound does not change from optimization in (5.3). Let us denote a point where $\mathcal{R}(P,Q)$ meets the blue line by the point $(1-\alpha'-\tau,1-\alpha')$ in the 2D plane, parametrized by $\alpha' \in [0,1-\tau]$. The $\mathcal{R}(\alpha',\tau)$ defined in Figure 6.2 works in this case also. We only need a new outer bound.

We construct an outer bound region, according to the following rule. We fit a hexagon where one edge is the 45-degree line passing through the origin, one edge is the vertical axis, one edge is the horizontal line passing through $(1,1)$, one edge is the 45-degree line with shift

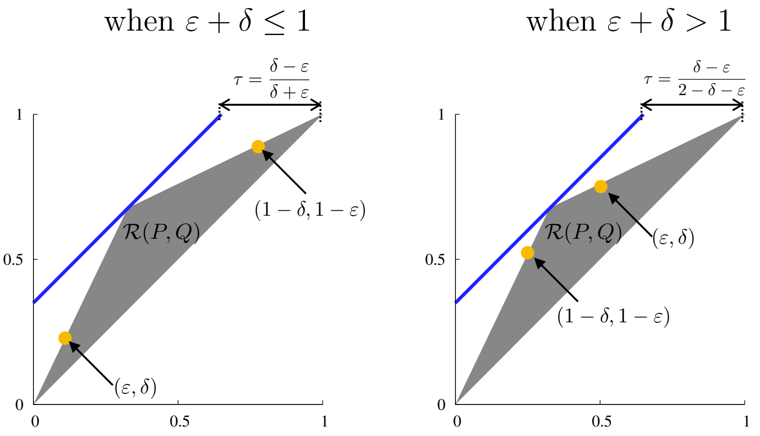

when $\varepsilon+\delta\le 1$
when $\varepsilon+\delta>1$
$\tau=\frac{\delta-\varepsilon}{\delta+\varepsilon}$
$\tau=\frac{\delta-\varepsilon}{2-\delta-\varepsilon}$
1
0.5
0
0
0.5
1
$\mathcal{R}(P,Q)$
$(1-\delta,1-\varepsilon)$
(ε,δ)
1
0.5
0
0
0.5
1
$\mathcal{R}(P,Q)$
(ε,δ)
$(1-\delta,1-\varepsilon)$

Figure 6.6: When $\delta+\varepsilon\le 1$ and $\tau=(\delta-\varepsilon)/(\delta+\varepsilon)$ (i.e. $(1-\tau)/2:(1+\tau)/2=\varepsilon:\delta$), a triangle mode collapse region that touches both points (ε,δ) and $(1-\delta,1-\varepsilon)$ at two of its edges also touches the 45-degree line with a τ shift at a vertex (left). When $\delta+\varepsilon>1$, the same happens when $\tau=(\delta-\varepsilon)/(2-\delta-\varepsilon)$ (i.e. $(1-\tau)/2:(1+\tau)/2=(1-\delta):(1-\varepsilon)$). Hence, if $\tau>\max\{(\delta-\varepsilon)/(\delta+\varepsilon),(\delta-\varepsilon)/(2-\delta-\varepsilon)\}$, then the triangle region that does not include both orange points cannot touch the blue 45-degree line.

τ shown in blue in Figure 6.7, and the remaining two edges include the two orange points, respectively, at (ε,δ) and $(1-\delta,1-\varepsilon)$. For any $\mathcal{R}(P,Q)$ satisfying the constraints in (5.13), there exists at least one such hexagon that includes $\mathcal{R}(P,Q)$. We parametrize the hexagon by α and β, where $(\alpha,\tau+\alpha)$ denotes the left-most point where the hexagon meets the blue line, and $(1-\tau-\beta,1-\beta)$ denotes the right-most point where the hexagon meets the blue line.

The additional constraint that (P,Q) has no (ε,δ)-mode collapse or (ε,δ)-mode augmentation translates into a geometric constraint that we need to consider all regions $\mathcal{R}(P,Q)$ that does not include the orange solid circle at point (ε,δ) and $(1-\delta,1-\varepsilon)$. Then, for any such (P,Q), we can sandwich the region $\mathcal{R}(P,Q)$ between two regions $\mathcal{R}_{\text{inner}}$ and $\mathcal{R}_{\text{outer1}}$:

$$\mathcal{R}_{\text{inner}}(\alpha',\tau)\ \subseteq\ \mathcal{R}(P,Q)\ \subseteq\ \mathcal{R}_{\text{outrer1}}(\varepsilon,\delta,\alpha,\beta,\tau)\ , \tag{6.7}$$

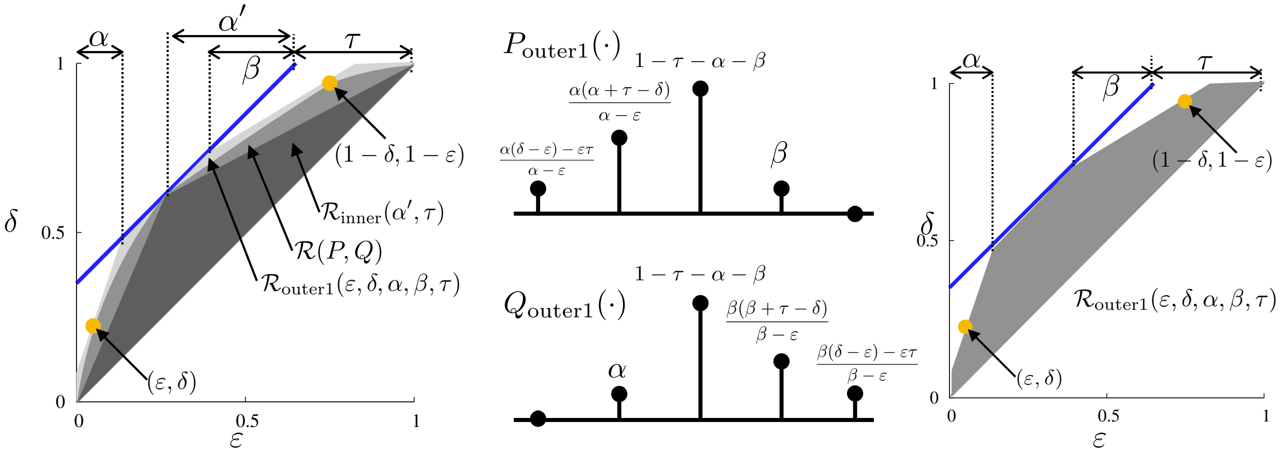
where $\mathcal{R}(\alpha,\tau)$ is defined as in Figure 6.2.

Let

$$(P_{\text{inner}}(\alpha'),Q_{\text{inner}}(\alpha',\tau))$$

defined in (5.5) and (5.6), and

$$(P_{\text{outer1}}(\varepsilon,\delta,\alpha,\beta,\tau),Q_{\text{outer1}}(\varepsilon,\delta,\alpha,\beta,\tau))$$



α
α'
β
τ
1
0.5
0
δ
ε
$(1-\delta, 1-\varepsilon)$
$\mathcal{R}_{\text{inner}}(\alpha', \tau)$
$\mathcal{R}(P, Q)$
$\mathcal{R}_{\text{outer1}}(\varepsilon, \delta, \alpha, \beta, \tau)$
(ε, δ)
$P_{\text{outer1}}(\cdot)$
$1-\tau-\alpha-\beta$
$\frac{\alpha(\alpha+\tau-\delta)}{\alpha-\varepsilon}$
$\frac{\alpha(\delta-\varepsilon)-\varepsilon\tau}{\alpha-\varepsilon}$
β
$Q_{\text{outer1}}(\cdot)$
$1-\tau-\alpha-\beta$
$\frac{\beta(\beta+\tau-\delta)}{\beta-\varepsilon}$
$\frac{\beta(\delta-\varepsilon)-\varepsilon\tau}{\beta-\varepsilon}$
α
α
β
τ
$(1-\delta, 1-\varepsilon)$
$\mathcal{R}_{\text{outer1}}(\varepsilon, \delta, \alpha, \beta, \tau)$
(ε, δ)

Figure 6.7: For any pair (P, Q) with no (ε, δ)-mode collapse or no (ε, δ)-mode augmentation, the corresponding region $\mathcal{R}(P, Q)$ is sandwiched between $\mathcal{R}_{\text{inner}}(\alpha', \tau)$ and $\mathcal{R}_{\text{outer1}}(\varepsilon, \delta, \alpha, \beta, \tau)$ (left). A canonical pair of distributions corresponding to $\mathcal{R}_{\text{outer1}}(\varepsilon, \delta, \alpha, \beta, \tau)$ (middle and right).

denote the pairs of canonical distributions achieving the inner and outer regions exactly as shown in Figure 6.7.

By the preservation of dominance under product distributions in Remark 5.5, it follows from the dominance in (6.7) that for any (P, Q) there exist α', α, and β such that

$$\begin{aligned}\mathcal{R}(P_{\text{inner}}(\alpha')^m, Q_{\text{inner}}(\alpha', \tau)^m) \;&\subseteq\; \mathcal{R}(P^m, Q^m) \;\subseteq \\ &\mathcal{R}(P_{\text{outer1}}(\varepsilon, \delta, \alpha, \beta, \tau)^m, Q_{\text{outer1}}(\varepsilon, \delta, \alpha, \beta, \tau)^m)\;. \end{aligned} \tag{6.8}$$

Due to the data processing inequality of mode collapse region in Remark 5.4, it follows that dominance of region implies dominance of total variation distances:

$$\begin{aligned}&\min_{\frac{\varepsilon\tau}{\delta-\varepsilon} \le \alpha' \le 1-\frac{\tau\delta}{\delta-\varepsilon}} d_{\text{TV}}(P_{\text{inner}}(\alpha')^m, Q_{\text{inner}}(\alpha', \tau)^m) \;\le\; d_{\text{TV}}(P^m, Q^m) \\ \le\; &\max_{\alpha, \beta \ge \frac{\varepsilon\tau}{\delta-\varepsilon}, \alpha+\beta \le 1-\tau} d_{\text{TV}}(P_{\text{outer1}}(\varepsilon, \delta, \alpha, \beta, \tau)^m, Q_{\text{outer1}}(\varepsilon, \delta, \alpha, \beta, \tau)^m)\;. \end{aligned} \tag{6.9}$$

The RHS and LHS of the above inequalities can be completely characterized by taking the m-th power of those canonical pairs of distributions, and then taking the respective minimum over α' and maximum over α and β. For the upper bound, this gives $U_1(\epsilon, \delta, \tau, m)$ in (5.14), and for the lower bound this gives $L_2(\tau, m)$ in (5.17).

Now, suppose $\delta + \varepsilon > 1$, and consider the intermediate regime when $\delta - \varepsilon \le \tau \le (\delta - \varepsilon)/(2 - \delta - \varepsilon)$. We have a different outer bound $\mathcal{R}_{\text{outer2}}(\varepsilon, \delta, \alpha, \delta, \tau)$ as the role of (ε, δ) and

$(1-\delta, 1-\varepsilon)$ have switched. A similar analysis gives

$$\begin{aligned} & d_{\mathrm{TV}}(P^m, Q^m) \\ \leq & \max_{\alpha,\beta \geq \frac{(1-\delta)\tau}{\delta-\varepsilon}, \alpha+\beta\leq 1-\tau} d_{\mathrm{TV}}(P_{\mathrm{outer2}}(\varepsilon,\delta,\alpha,\beta,\tau)^m, Q_{\mathrm{outer2}}(\varepsilon,\delta,\alpha,\beta,\tau)^m) \, , \end{aligned} \tag{6.10}$$

where the canonical distributions are shown in Figure 6.8 and defined in (5.19) and (5.20). This gives $U_2(\epsilon, \delta, \tau, m)$ in (5.18). For the lower bound we only need to change the range of α we minimize over, which gives $L_3(\tau, m)$ in (5.21).

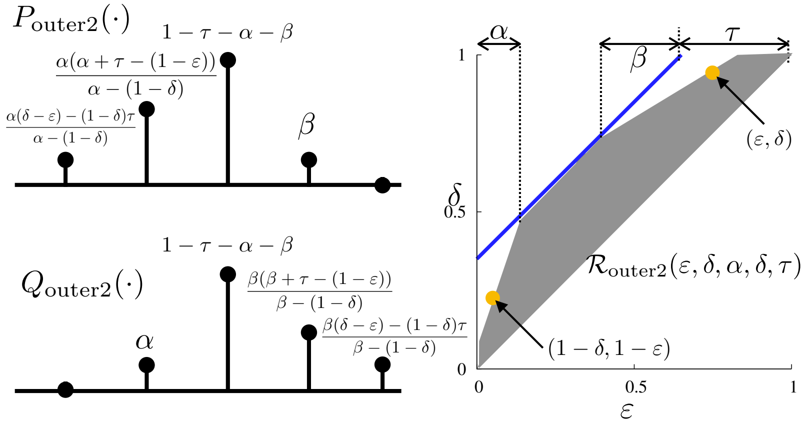

$P_{\mathrm{outer2}}(\cdot)$
$1-\tau-\alpha-\beta$
$\frac{\alpha(\alpha+\tau-(1-\varepsilon))}{\alpha-(1-\delta)}$
$\frac{\alpha(\delta-\varepsilon)-(1-\delta)\tau}{\alpha-(1-\delta)}$
β
$1-\tau-\alpha-\beta$
$Q_{\mathrm{outer2}}(\cdot)$
$\frac{\beta(\beta+\tau-(1-\varepsilon))}{\beta-(1-\delta)}$
$\frac{\beta(\delta-\varepsilon)-(1-\delta)\tau}{\beta-(1-\delta)}$
α
α
β
τ
1
δ
0.5
0
(ε, δ)
$\mathcal{R}_{\mathrm{outer2}}(\varepsilon, \delta, \alpha, \delta, \tau)$
$(1-\delta, 1-\varepsilon)$
0
0.5
1
ε

Figure 6.8: A canonical pair of distributions corresponding to $\mathcal{R}_{\mathrm{outer2}}(\varepsilon, \delta, \alpha, \beta, \tau)$.

REFERENCES

[1] D. Blackwell, "Equivalent comparisons of experiments," *The Annals of Mathematical Statistics*, vol. 24, no. 2, pp. 265–272, 1953.

[2] A. Srivastava, L. Valkov, C. Russell, M. Gutmann, and C. Sutton, "Veegan: Reducing mode collapse in gans using implicit variational learning," *arXiv preprint arXiv:1705.07761*, 2017.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[5] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.

[6] E. L. Denton, S. Chintala, R. Fergus et al., "Deep generative image models using a laplacian pyramid of adversarial networks," in *Advances in neural information processing systems*, 2015, pp. 1486–1494.

[7] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient." in *AAAI*, 2017, pp. 2852–2858.

[8] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 613–621.

[9] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, and Z. Wang, "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint arXiv:1609.04802*, 2016.

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint arXiv:1611.07004*, 2016.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[12] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3730–3738.

[13] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[14] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[15] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016.

[16] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *arXiv preprint arXiv:1605.09782*, 2016.

[17] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville, "Adversarially learned inference," *arXiv preprint arXiv:1606.00704*, 2016.

[18] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," *arXiv preprint arXiv:1611.02163*, 2016.

[19] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," *arXiv preprint arXiv:1612.02136*, 2016.

[20] Y. Saatci and A. Wilson, "Bayesian gans," in *Advances in Neural Information Processing Systems*, 2017, pp. 3624–3633.

[21] T. Nguyen, T. Le, H. Vu, and D. Phung, "Dual discriminator generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2017, pp. 2667–2677.

[22] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[23] A. Stam, "Some inequalities satisfied by the quantities of information of fisher and shannon," *Information and Control*, vol. 2, no. 2, pp. 101–112, 1959.

[24] A. Dembo, T. M. Cover, and J. A. Thomas, "Information theoretic inequalities," *Information Theory, IEEE Transactions on*, vol. 37, no. 6, pp. 1501–1518, 1991.

[25] T. M. Cover and A. Thomas, "Determinant inequalities via information theory," *SIAM journal on Matrix Analysis and Applications*, vol. 9, no. 3, pp. 384–392, 1988.

[26] R. Zamir, "A proof of the fisher information inequality via a data processing argument," *Information Theory, IEEE Transactions on*, vol. 44, no. 3, pp. 1246–1250, 1998.

[27] S. Verdú and D. Guo, "A simple proof of the entropy-power inequality," *IEEE Transactions on Information Theory*, vol. 52, no. 5, pp. 2165–2166, 2006.

[28] T. Liu and P. Viswanath, "An extremal inequality motivated by multiterminal information-theoretic problems," *Information Theory, IEEE Transactions on*, vol. 53, no. 5, pp. 1839–1851, 2007.

[29] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 2879–2887.

[30] P. Kairouz, S. Oh, and P. Viswanath, "Secure multi-party differential privacy," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[31] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," *IEEE Transactions on Information Theory*, vol. 63, no. 6, pp. 4037–4049, June 2017.

[32] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[33] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and equilibrium in generative adversarial nets (GANs)," *arXiv preprint arXiv:1703.00573*, 2017.

[34] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[35] L. Theis, A. v. d. Oord, and M. Bethge, "A note on the evaluation of generative models," *arXiv preprint arXiv:1511.01844*, 2015.

[36] Y. Wu, Y. Burda, R. Salakhutdinov, and R. Grosse, "On the quantitative analysis of decoder-based generative models," *arXiv preprint arXiv:1611.04273*, 2016.

[37] S. Santurkar, L. Schmidt, and A. Madry, "A classification-based perspective on gan distributions," *arXiv preprint arXiv:1711.00970*, 2017.

[38] S. Arora and Y. Zhang, "Do gans actually learn the distribution? an empirical study," *arXiv preprint arXiv:1706.08224*, 2017.

[39] I. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Schölkopf, "Adagan: Boosting generative models," *arXiv preprint arXiv:1701.02386*, 2017.

[40] J. Li, A. Madry, J. Peebles, and L. Schmidt, "Towards understanding the dynamics of generative adversarial networks," *arXiv preprint arXiv:1706.09884*, 2017.

[41] S. Liu, O. Bousquet, and K. Chaudhuri, "Approximation and convergence properties of generative adversarial learning," *arXiv preprint arXiv:1705.08991*, 2017.

[42] B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. Lanckriet, "Hilbert space embeddings and metrics on probability measures," *Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1517–1561, 2010.

[43] S. Feizi, C. Suh, F. Xia, and D. Tse, "Understanding gans: the lqg setting," *arXiv preprint arXiv:1710.10793*, 2017.

[44] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[46] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[47] K. Mills and I. Tamblyn, “Phase space sampling and operator confidence with generative adversarial networks,” *arXiv preprint arXiv:1710.08053*, 2017.

[48] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, “Compressed sensing using generative models,” *arXiv preprint arXiv:1703.03208*, 2017.