PacGAN: The Power of Two Samples in Generative Adversarial Networks

Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh, *Member, IEEE,*

***Abstract*—Generative adversarial networks (GANs) are innovative techniques for learning generative models of complex data distributions from samples. Despite remarkable 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 recently-proposed approaches mitigate mode collapse. We propose a principled approach to handle mode collapse called *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.**

***Index Terms*—generative adversarial networks, mode collapse, hypothesis testing, data processing inequalities**

I. Introduction

Generative adversarial networks (GANs) are an innovative technique for training generative models to produce realistic examples from a data distribution [2]. 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 [3], [4], [2]. 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 by introducing an innovative minimax formulation whose solution is approximated by iteratively training two competing neural networks.

GANs have attracted a great deal of interest recently. They are able to *generate* realistic, crisp, and original examples of images [2], [5] and text [6]. This is useful in image and video processing (e.g. frame prediction [7], image super-resolution [8], and image-to-image translation [9]), 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 [10]). GANs may be able to provide an unsupervised approach to learning representations that capture semantics of arbitrary data structures and applications, for downstream tasks like image manipulation [11] and defending against adversarial examples [12].

a) 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, a primary goal is to train such a generator from unlabelled training data drawn independently from a distribution (e.g., faces [13] or natural images [14]), to produce realistic samples that are different from the training data.

GANs achieved a breakthrough in training such generative models [2]. 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. Concretely, we search for (the parameters of) neural networks G and D that optimize the following minimax objective:

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

where P is the distribution of the real data, and P_Z is the

Zinan Lin is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, GA, 15213 USA e-mail: zinanl@andrew.cmu.edu.

Ashish Khetan is with Amazon e-mail: ashish.khetan09@gmail.com.

Giulia Fanti is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, GA, 15213 USA e-mail: gfanti@andrew.cmu.edu.

Sewoong Oh is with the Allen School of Computer Science and Engineering, University of Washington, Seattle, WA, 98195 USA e-mail: sewoong@cs.washington.edu.

distribution of the input random 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, [2] shows that the global optimum of (1) is achieved if and only if $P = Q$, where Q is the generated distribution of $G(Z)$ under some mild assumptions. Section III discusses this minimax formulation in detail. The solution to optimization (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.

b) 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 [15]. Similarly, in tasks that translate a caption into an image, generators have been shown to generate series of nearly-identical images [16]. 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. [17], [18], [19], [15], [20], [21], [22], [23]. Proposed solutions rely on modified architectures [17], [18], [19], [15], loss functions [21], [24], and optimization algorithms [20]. 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 of their empirical gains, especially when those gains are sensitive to architecture hyperparameters.

c) 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 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 Section III-A). This definition is closely related to the notions of false alarm and missed detection in binary hypothesis testing (see Section III-C). 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 (Section III-A).

(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 Section III-B). 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 sharp results with simple proofs. This follows a long tradition in information theory literature (e.g. [25], [26], [27], [28], [29], [30], [31], [32], [33]) 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 Section II). 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 Section III-B). We demonstrate on benchmark datasets that PacGAN significantly improves upon competing approaches in mitigating mode collapse. Further, unlike existing approaches on jointly using multiple samples, e.g. [15], PacGAN requires no hyper-parameter tuning and incurs only a slight overhead in the architecture.

d) Outline: This paper is structured as follows: we present the PacGAN framework in Section II. In Section III, we propose a *new* definition of mode collapse, and provide analyses showing that PacGAN mitigates mode collapse. We refer to a longer version of this paper [34] for proofs of the main results, detailed discussion of the related work, and extensive experimental results, demonstrating that we significantly improve over competing state-of-the-art schemes designed to mitigate mode collapse on all benchmark datasets.

II. PacGAN: A 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

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

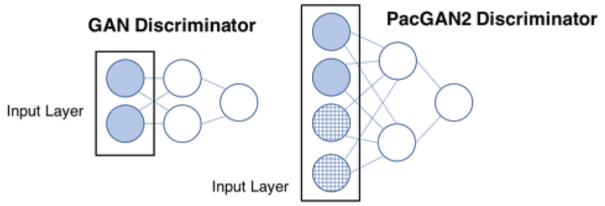

Fig. 1. PacGAN(m) augments the input layer by a factor of m. The number of edges between the first two layers is increased accordingly to preserve the connectivity of the mother architecture (e.g., fully-connected). Packed samples are fed to the input layer in a concatenated fashion; the gridded nodes represent input nodes for the second input sample.

$Y = 0$). 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 [2] and feed the discriminator three packed samples as input, we call this "PacGAN3". If we take the celebrated DCGAN [35] and feed the discriminator four packed samples as input, we call this "PacDCGAN4". We use PacGAN without a subsequent integer to refer to the principle of packing.

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 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 gridded nodes in Figure 1 represent input nodes for the second sample.

Similarly, when packing a DCGAN, which uses (de-)convolutional neural networks for both the generator and the discriminator, we simply stack the images along the color channel. For instance, the discriminator for PacDCGAN5 on the MNIST dataset of handwritten images [36] would take an input of size $28 \times 28 \times 5$, since each individual black-and-white MNIST image is 28×28 pixels; a discriminator for PacDCGAN5 on the $32 \times 32 \times 5$ CelebA dataset [13] would take an input of size $32 \times 32 \times 15$. Only the input layer and the number of weights in the corresponding first convolutional layer will increase in depth by a factor of five for these two examples. 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 experiments.

How to train a packed discriminator. The only difference between the training of PacGAN and standard GANs is that each minibatch in PacGAN contains *packed* samples instead of single samples. More precisely, each sample in the minibatch 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 Section III.

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 [18], [17], [15], [19], [20].

a) 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 $\mathcal{O}(g^2)$ per input sample, assuming a naive implementation. Hence the overall minibatch update complexity is $\mathcal{O}(rwg^2)$. 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 $\mathcal{O}((w + m)rg^2)$. We find that in practice, even $m = 2$ or $m = 3$ give good results. Also note that the overhead can be much less in practice with GPUs, which can parallelize these additional computations.

III. Theoretical analyses of PacGAN

In this section, 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*, which is motivated by the ROC (Receiver Operating Characteristic) curve representation of binary hypothesis testing.

Mode collapse is a phenomenon commonly reported in the GAN literature [37], [16], [38], [39], [40], 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 [37]. 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 [37]. In this paper, 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:

$$\min_G \quad \underbrace{\max_D \;\; \mathbb{E}_{X\sim P}[\log(D(X))] + \mathbb{E}_{G(Z)\sim Q}[\log(1-D(G(Z)))]}_{\simeq\; d_{\rm KL}\left(P\|\frac{P+Q}{2}\right)+d_{\rm KL}\left(Q\|\frac{P+Q}{2}\right)+\log(1/4)} \;, \qquad (2)$$

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 (up to a scaling and shift by a constant), defined as $d_{\rm JS}(P,Q) \triangleq (1/2)\, d_{\rm KL}(P\|(P+Q)/2) + (1/2)\, d_{\rm KL}(Q\|(P+Q)/2)$, where $d_{\rm KL}(\cdot)$ 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 $D(X) = P(X)/(P(X)+Q(X))$ [2]. 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 section, 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:

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

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. [41], [42]). 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 3, 4, and 5. The last assumption gives a loss of the total variation distance $d_{\rm TV}(P,Q) \triangleq \sup_{S\subseteq\mathcal{X}}\{P(S)-Q(S)\}$ over the domain $\mathcal{X}$. This follows from the fact that (e.g. [37]),

$$\begin{aligned} & \sup_D \left\{\mathbb{E}_{X\sim P}[\mathbb{I}(D(X))] + \mathbb{E}_{G(Z)\sim Q}[1-\mathbb{I}(D(G(Z)))]\right\} \\ = & \sup_S \left\{P(S) + 1 - Q(S)\right\} \\ = & \; 1 + d_{\rm TV}(P,Q) \;. \end{aligned}$$

This discriminator provides (an approximation of) the total variation distance, and the generator tries to minimize the total variation distance $d_{\rm TV}(P,Q)$. 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 Section III-B. 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 $d_{\rm TV}(P^m, Q^m)$.

We first provide a formal mathematical definition of mode collapse in Section III-A, 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 $d_{\rm TV}(P^m,Q^m)$ seen by a packed discriminator (Section III-B). The proofs of these results are provided in [34]. In Section III-C, we show that the proposed mode collapse region is equivalent to the

ROC curve for binary hypothesis testing. This allows us to use powerful mathematical techniques from binary hypothesis testing including the data processing inequality and the reverse data processing inequalities.

A. Mathematical definition of mode collapse as a 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. A major challenge with this definition 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 the geometric constraint, and introduce a purely analytical definition.

Definition 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.* Consider a toy example in Figure 2, 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 2. They are designed to have the same total variations distance, i.e. $d_{\rm TV}(P,Q_1) = d_{\rm TV}(P,Q_2) = 0.2$, 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 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 2.

$$\mathcal{R}(P,Q) \triangleq \\ \mathrm{conv}\Big(\big\{ (\varepsilon,\delta) \,\big|\, \delta > \varepsilon \text{ and } (P,Q) \text{ has } (\varepsilon,\delta)\text{-mode collapse} \big\} \Big) , \tag{3}$$

where $\mathrm{conv}(\cdot)$ denotes the convex hull. Using the convex hull in this definition makes sure that the same definition seamlessly interpolates between continuous and discrete variables. In particular, for continuous variables, the original region is convex and we do not need to take a 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 for the purpose of hypothesis testing. This assertion is made precise in Section III-C 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. 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 2 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 $(0,0)$ of the upper boundary of the shaded grey region (in this example the slope is in fact infinite). The right panel in Figure 2 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 $(0,0)$ of the upper boundary of the shaded grey region.

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 2. *A target distribution P and a generator Q have (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 $P(S) \leq \varepsilon$.*

We distinguish mode collapse and augmentation strictly for analytical purposes. In GAN literature, both collapse and augmentation contribute to the "mode collapse" phenomenon.

B. Evolution of the region under product distributions

The toy example generators Q_1 and Q_2 from Figure 2 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. We prove in Theorem 4 that a discriminator that packs multiple samples together *can* better distinguish mode-collapsing generators. Intuitively, m packed samples are equivalent to a single sample 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 $d_{\rm TV}(P^m,Q^m)$.

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 3 illustrates how the mode

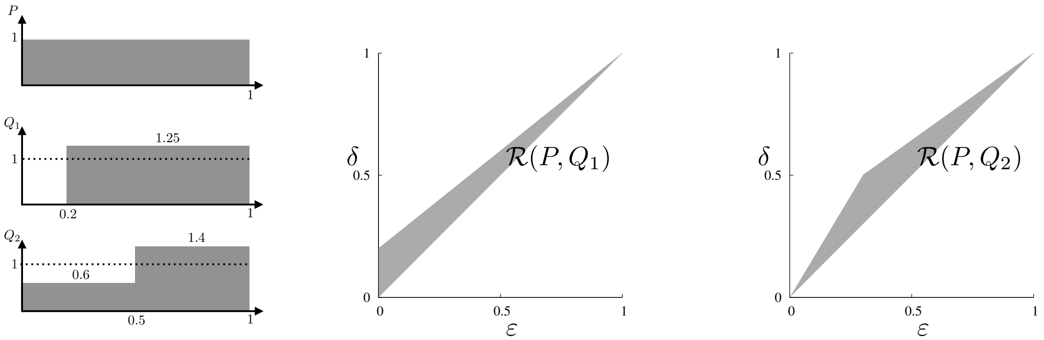

Fig. 2. A formal definition of (ε, δ)-mode collapse and its accompanying region representation captures the intensity of mode collapse for generators Q_1 with mode collapse and Q_2 which does not have mode collapse, for a toy example distributions P, Q_1, and Q_2 shown on the left. The region of (ε, δ)-mode collapse that is achievable is shown in grey.

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. In particular, a larger mode collapse region implies a larger total variation distance between P and the generator. The total variation distances $d_{\text{TV}}(P^m, Q_1^m)$ and $d_{\text{TV}}(P^m, Q_2^m)$, 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.

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:

$$\begin{aligned} &\min / \max_{P,Q} \;\; d_{\text{TV}}(P^m, Q^m) \\ &\text{subject to } d_{\text{TV}}(P, Q) = \tau \; , \end{aligned} \tag{4}$$

where the maximization and minimization are over all probability measures P and Q. We give the exact solution in Theorem 3, which is illustrated pictorially in Figure 4 (left).

Theorem 3. *For all $0 \le \tau \le 1$ and a positive integer m, the solution to the maximization in* (4) *is $1 - (1 - \tau)^m$, and the solution to the minimization in* (4) *is*

$$L(\tau, m) \triangleq \min_{0 \le \alpha \le 1-\tau} \; d_{\text{TV}}\Big(P_{\text{inner}}(\alpha)^m, Q_{\text{inner}}(\alpha, \tau)^m \Big) \; , \tag{5}$$

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

$$P_{\text{inner}}(\alpha) \;=\; \big[1 - \alpha, \quad \alpha\big] \; , \tag{6}$$

$$Q_{\text{inner}}(\alpha, \tau) \;=\; \big[1 - \alpha - \tau, \quad \alpha + \tau\big] \; . \tag{7}$$

Although this is a simple statement that can be proved in several different ways, we introduce 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 4, for which other techniques do not generalize. Note that the claim in Theorem 3 has nothing to do with mode collapse. Still, the mode collapse region definition (used here purely as a proof technique) provides a novel technique that seamlessly generalizes to prove more complex statements in the following.

For any given value of τ and m, the bounds in Theorem 3 are easy to evaluate numerically, as shown below in the left panel of Figure 4. Within this achievable range, some subset of pairs (P, Q) have rapidly increasing total variation, occupying the upper part of the region (shown in red, middle panel of Figure 4), and some subset of pairs (P, Q) have slowly increasing total variation, occupying the lower part as shown in blue in the right panel in Figure 4. 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.

Evolution of total variation distances with mode collapse. We analyze how the total variation evolves for the set of all

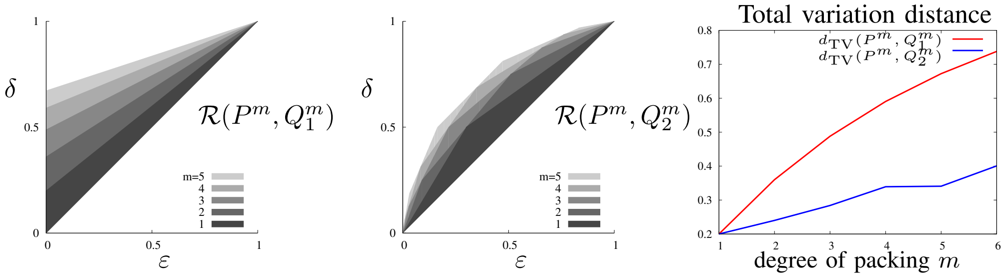

Fig. 3. Evolution of the mode collapse region over the degree of packing m for the two toy examples from Figure 2. The region of the mode-collapsing generator Q_1 expands faster than the non-mode-collapsing generator Q_2 when discriminator inputs are packed (at $m = 1$ these examples have the same TV distances). This causes a discriminator to penalize mode collapse as desired.

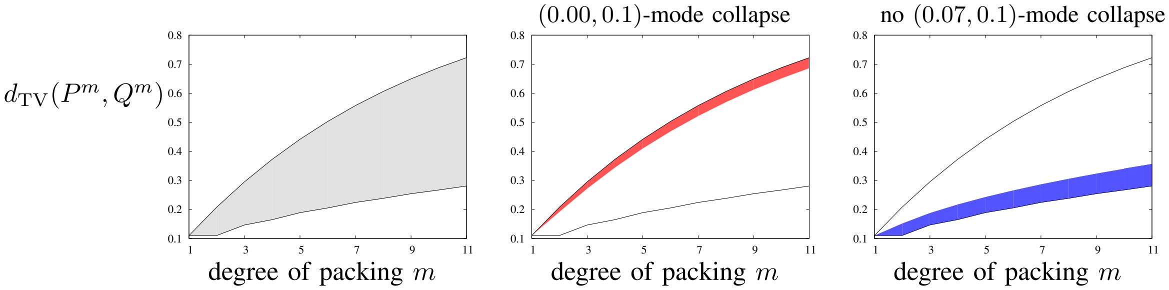

Fig. 4. The range of $d_{\mathrm{TV}}(P^m, Q^m)$ achievable by pairs with $d_{\mathrm{TV}}(P, Q) = \tau$, for a choice of $\tau = 0.11$, defined by the solutions of the optimization (4) provided in Theorem 3 (left panel). The range of $d_{\mathrm{TV}}(P^m, Q^m)$ achievable by those pairs that also have $(\varepsilon = 0.00, \delta = 0.1)$-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).

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 gives the desired range of total variation distances:

$$\begin{aligned} & \min / \max_{P,Q} \;\; d_{\mathrm{TV}}(P^m, Q^m) \\ & \text{subject to} \;\; d_{\mathrm{TV}}(P, Q) = \tau \\ & (P, Q) \text{ has } (\varepsilon, \delta)\text{-mode collapse} \; , \end{aligned} \tag{8}$$

where the maximization and minimization are over all probability measures P and Q, and the mode collapse constraint is defined in Definition 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 shows that mode-collapsing pairs occupy the upper part of the total variation region; that is, total variation increases rapidly with packing (Figure 4, middle). One implication is that distribution pairs (P, Q) at the top of the total variation evolution region exhibit the strongest mode collapse. Also, a pair (P, Q) with strong mode collapse (i.e., with larger δ and smaller ε in the constraint) will be penalized more by packing; hence, a generator minimizing an approximation of $d_{\mathrm{TV}}(P^m, Q^m)$ will be unlikely to select a distribution with strong mode collapse.

Theorem 4. *For all $0 \leq \varepsilon < \delta \leq 1$ and a positive integer m, if $1 \geq \tau \geq \delta - \varepsilon$ the solution to the maximization in* (8) *is $1 - (1 - \tau)^m$, and the solution to the minimization in* (8) *is*

$$\begin{aligned} & L_1(\varepsilon, \delta, \tau, m) \triangleq \\ & \min \Bigg\{ \min_{0 \leq \alpha \leq 1 - \frac{\tau\delta}{\delta - \varepsilon}} d_{\mathrm{TV}}\Big(P_{\mathrm{inner1}}(\delta, \alpha)^m, Q_{\mathrm{inner1}}(\varepsilon, \alpha, \tau)^m \Big) \, , \\ & \min_{1 - \frac{\tau\delta}{\delta - \varepsilon} \leq \alpha \leq 1 - \tau} d_{\mathrm{TV}}\Big(P_{\mathrm{inner2}}(\alpha)^m, Q_{\mathrm{inner2}}(\alpha, \tau)^m \Big) \Bigg\} \, , \end{aligned} \tag{9}$$

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

$$P_{\mathrm{inner1}}(\delta, \alpha) = \Big[\delta, \;\; 1 - \alpha - \delta, \;\; \alpha \Big] \, , \tag{10}$$

$$Q_{\mathrm{inner1}}(\varepsilon, \alpha, \tau) = \Big[\varepsilon, \;\; 1 - \alpha - \tau - \varepsilon, \;\; \alpha + \tau \Big] \, , \tag{11}$$

$$P_{\mathrm{inner2}}(\alpha) = \Big[1 - \alpha, \;\; \alpha \Big] \, , \tag{12}$$

$$Q_{\mathrm{inner2}}(\alpha, \tau) = \Big[1 - \alpha - \tau, \;\; \alpha + \tau \Big] \, . \tag{13}$$

If $\tau < \delta - \varepsilon$, the optimization in (8) *has no solution and the feasible set is an empty set.*

The proof relies on the mode collapse region representation of the pair (P, Q) and Blackwell's result [1]. The solutions in Theorem 4 can be numerically evaluated for a given $(\varepsilon, \delta, \tau)$ as in Figure 5. Analogous results can be shown for pairs (P, Q) that exhibit (ϵ, δ) mode augmentation as straightforward extensions of the mode collapse proofs. This holds because TV distance is a metric, and therefore symmetric.

Evolution of total variation distances without mode collapse. We next analyze how total variation evolves for the set of all pairs (P, Q) that have the same (unpacked) total

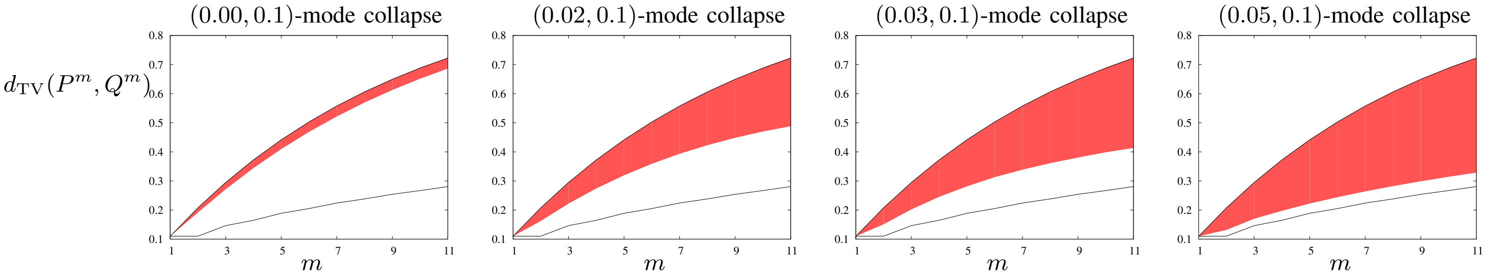

Fig. 5. The evolution of total variation distance over the packing degree m for mode collapsing pairs is shown as a red band. The upper and lower boundaries of the red band is defined by the optimization 8 and computed using Theorem 4. For a fixed $d_{\mathrm{TV}}(P, Q) = \tau = 0.11$ and $(\varepsilon, \delta = 0.1)$-mode collapse, we show the evolution with different choices of $\varepsilon \in \{0.00, 0.02, 0.03, 0.05\}$. The black solid lines show the maximum/minimum total variation in optimization (4) 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.

variation distances τ and *do not* have (ε, δ)-mode collapse for some $0 \le \varepsilon < \delta \le 1$. By the symmetry of total variation distance, mode augmentation in Definition 2 is as damaging as mode collapse in terms of evolution of total variation distance. Hence, we characterize this evolution for families of distribution pairs without either mode collapse or augmentation. The following optimization problem gives the desired range of total variation distances:

$$\begin{aligned} &\min / \max_{P,Q} \;\; d_{\mathrm{TV}}(P^m, Q^m) &(14)\\ &\text{subject to} \;\;\; d_{\mathrm{TV}}(P, Q) = \tau \\ &(P, Q) \text{ does not have } (\varepsilon, \delta)\text{-mode collapse/augmentation} \;, \end{aligned}$$

where the maximization and minimization are over all probability measures P and Q, and the mode collapse and augmentation constraints are defined in Definitions 1 and 2, respectively.

It is not possible to have $d_{\mathrm{TV}}(P, Q) > (\delta - \varepsilon)/(\delta + \varepsilon)$ and $\delta + \varepsilon \le 1$, and satisfy the mode collapse and mode augmentation constraints. Similarly, it is not possible to have $d_{\mathrm{TV}}(P, Q) > (\delta - \varepsilon)/(2 - \delta - \varepsilon)$ 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 (4) with no mode collapse constraints. A 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 in equation (18) is the same as the lower bound in (5), 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; 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 (Figure 4 right), and will be penalized less under packing. Hence a generator minimizing (approximate) $d_{\mathrm{TV}}(P^m, Q^m)$ is likely to generate distributions with weak mode collapse.

Theorem 5. *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* (14) *are the same as those of the optimization* (4) *provided in Theorem 3.*

If $\delta + \varepsilon \le 1$ and $\delta - \varepsilon \le \tau \le (\delta - \varepsilon)/(\delta + \varepsilon)$ then the solution to optimization (14) *is*

$$\begin{aligned} &U_1(\epsilon, \delta, \tau, m) \triangleq \\ &\max_{\alpha+\beta \le 1-\tau,\, \frac{\varepsilon\tau}{\delta-\varepsilon} \le \alpha, \beta} \; d_{\mathrm{TV}}\Big(P_{\mathrm{outer1}}(\varepsilon, \delta, \alpha, \beta, \tau)^m, \\ &\qquad\qquad Q_{\mathrm{outer1}}(\varepsilon, \delta, \alpha, \beta, \tau)^m \Big) \;, \qquad (15) \end{aligned}$$

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

$$\begin{aligned} &P_{\mathrm{outer1}}(\varepsilon, \delta, \alpha, \beta, \tau) = &(16)\\ &\left[\tfrac{\alpha(\delta-\varepsilon)-\varepsilon\tau}{\alpha-\varepsilon}, \;\; \tfrac{\alpha(\alpha+\tau-\delta)}{\alpha-\varepsilon}, \;\; 1-\tau-\alpha-\beta, \;\; \beta, \;\; 0 \right], \text{ and} \\ &Q_{\mathrm{outer1}}(\varepsilon, \delta, \alpha, \beta, \tau) = &(17)\\ &\left[0, \;\; \alpha, \;\; 1-\tau-\alpha-\beta, \;\; \tfrac{\beta(\beta+\tau-\delta)}{\beta-\varepsilon}, \;\; \tfrac{\beta(\delta-\varepsilon)-\varepsilon\tau}{\beta-\varepsilon} \right] . \end{aligned}$$

The solution to the minimization in (14) *is*

$$\begin{aligned} &L_2(\tau, m) \triangleq &(18)\\ &\min_{\frac{\varepsilon\tau}{\delta-\varepsilon} \le \alpha \le 1-\frac{\delta\tau}{\delta-\varepsilon}} \; d_{\mathrm{TV}}\Big(P_{\mathrm{inner}}(\alpha)^m, Q_{\mathrm{inner}}(\alpha, \tau)^m \Big) \;, \end{aligned}$$

where $P_{\mathrm{inner}}(\alpha)$ and $Q_{\mathrm{inner}}(\alpha, \tau)$ are defined as in Theorem 3.

If $\delta + \varepsilon > 1$ and $\delta - \varepsilon \le \tau \le (\delta - \varepsilon)/(2 - \delta - \varepsilon)$ the solution to the maximization (14) *is*

$$\begin{aligned} &U_2(\epsilon, \delta, \tau, m) \triangleq \\ &\max_{\alpha+\beta \le 1-\tau,\, \frac{(1-\delta)\tau}{\delta-\varepsilon} \le \alpha, \beta} \; d_{\mathrm{TV}}\Big(P_{\mathrm{outer2}}(\varepsilon, \delta, \alpha, \beta, \tau)^m, \\ &\qquad\qquad Q_{\mathrm{outer2}}(\varepsilon, \delta, \alpha, \beta, \tau)^m \Big) \;, \qquad (19) \end{aligned}$$

where $P_{\mathrm{outer2}}(\varepsilon, \delta, \alpha, \beta, \tau)^m$ and $Q_{\mathrm{outer2}}(\varepsilon, \delta, \alpha, \beta, \tau)^m$ are the m-th order product distributions of discrete random variables distributed as

$$\begin{aligned} &P_{\mathrm{outer2}}(\varepsilon, \delta, \alpha, \beta, \tau) = &(20)\\ &\left[\tfrac{\alpha(\delta-\varepsilon)-(1-\delta)\tau}{\alpha-(1-\delta)}, \;\; \tfrac{\alpha(\alpha+\tau-(1-\varepsilon))}{\alpha-(1-\delta)}, \;\; 1-\tau-\alpha-\beta, \;\; \beta, \;\; 0 \right], \end{aligned}$$

and

$$\begin{aligned} &Q_{\mathrm{outer2}}(\varepsilon, \delta, \alpha, \beta, \tau) = &(21)\\ &\left[0, \;\; \alpha, \;\; 1-\tau-\alpha-\beta, \;\; \tfrac{\beta(\beta+\tau-(1-\varepsilon))}{\beta-(1-\delta)}, \;\; \tfrac{\beta(\delta-\varepsilon)-(1-\delta)\tau}{\beta-(1-\delta)} \right] . \end{aligned}$$

The solution to the minimization in (14) *is*

$$L_3(\tau, m) \triangleq \min_{\frac{(1-\delta)\tau}{\delta-\varepsilon} \le \alpha \le 1 - \frac{(1-\varepsilon)\tau}{\delta-\varepsilon}} d_{\mathrm{TV}}\Big(P_{\mathrm{inner}}(\alpha)^m, Q_{\mathrm{inner}}(\alpha, \tau)^m \Big) \,, \tag{22}$$

where $P_{\mathrm{inner}}(\alpha)$ and $Q_{\mathrm{inner}}(\alpha, \tau)$ are defined as in Theorem 3.

If $\tau > \max\{(\delta-\varepsilon)/(\delta+\varepsilon), (\delta-\varepsilon)/(2-\delta-\varepsilon)\}$, then the optimization in (14) *has no solution and the feasible set is an empty set.*

A proof of this theorem 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 can be numerically evaluated for any given choices of $(\varepsilon, \delta, \tau)$ as we show in Figure 6.

C. Interpretation of mode collapse via hypothesis testing regions

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.

1) Equivalence between the mode collapse region and the ROC curve: There is a simple one-to-one correspondence between mode collapse region as we define it in Section III-A (e.g. Figure 2) and the ROC curve 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 used to prove strong composition theorems in differential privacy [33] and to identify the optimal differentially-private mechanisms under local privacy [31] and multi-party communications [32]. Concretely, an ROC curve of a binary hypothesis testing is obtained by plotting the largest achievable true positive rate (TPR), i.e. $1-$probability of missed detection, or equivalently $1-$ type II error, on the vertical axis against the false positive rate (FPR), i.e probability of false alarm or equivalently type I error, on the horizontal axis.

We map the binary hypothesis testing setup directly to the GAN context. Suppose the null hypothesis $h = 0$ denotes observations drawn from the generated distribution Q, and the alternate hypothesis $h = 1$ denotes observations drawn from the true distribution P. Given a sample X, suppose we decide whether the sample came from P or Q based on a rejection region S_{reject}; i.e., we reject the null hypothesis if $X \in S_{\mathrm{reject}}$. FPR (i.e. Type I error) is when the null hypothesis is true but rejected, which happens with $\mathbb{P}(X \in S_{\mathrm{reject}} | h = 0)$, and TPR (i.e. 1-type II error) is when the null hypothesis is false and rejected, which happens with $\mathbb{P}(X \in S_{\mathrm{reject}} | h = 1)$. Sweeping through the achievable pairs $(\mathbb{P}(X \in S_{\mathrm{reject}} | h = 1), \mathbb{P}(X \in S_{\mathrm{reject}} | h = 0))$ for all possible rejection sets defines a two-dimensional, convex *hypothesis testing region*. The upper boundary of this convex set is the ROC curve. Example ROC curves for the two toy examples (P, Q_1) and (P, Q_2) from Figure 2 are shown in Figure 7.

In defining the region, we allow stochastic decisions; if two points (x, y) and (x', y') have achievable TPR and FPR, then any convex combination of those points is achievable by randomly choosing between the rejection sets. Hence, the resulting hypothesis testing region is alway convex by definition. We show only the region above the 45-degree line passing through $(0, 0)$ and $(1, 1)$, as the other region is symmetric and redundant. For a given pair (P, Q), a simple relation relates its mode collapse region and hypothesis testing region.

Remark 6 (Equivalence)**.** *For a pair of target P and generator Q, the hypothesis testing region is the same as the mode collapse region.*

This follows immediately from the definition of mode collapse region (Definition 1). If there exists a set S such that $P(S) = \delta$ and $Q(S) = \varepsilon$, then for the choice of $S_{\mathrm{reject}} = S$ in the binary hypothesis test, the point $(\mathbb{P}(X \in S_{\mathrm{reject}} | h = 0) = \varepsilon, \mathbb{P}(X \in S_{\mathrm{reject}} | h = 1) = \delta)$ in the hypothesis testing region is achievable. The converse is also true, if we make deterministic decisions on S_{reject}. As the mode collapse region is a convex hull of all achievable points, points in the hypothesis testing region requiring randomized decisions can also be covered. For example, the hypothesis testing regions of the toy examples from Figure 2 are shown below in Figure 7. This simple relation allows us to use rich analysis tools known for hypothesis testing regions and ROC curves.

IV. Experiments

Due to space constraints, we defer the bulk of our empirical results to the longer version of this paper in [34]. However, we include here some basic empirical results on synthetic and real datasets. We start with a visual demonstration of PacGAN's efficacy on a toy dataset called 2-D Grid, consisting of a Gaussian mixture of 25 modes in a grid arrangement (Figure 8, left). Figure 8 shows the modes learned by a vanilla GAN (center) and by PacGAN2 (right). We observe that while vanilla GANs miss several modes, PacGAN2 is able to recover all of them.

To quantify this effect, we measure both the number of modes captured and the sample quality on the 2D-Grid, as well as a related dataset, the 2D-Ring (an 8-mode mixture of 2D Gaussians arranged in a ring). On these standard Gaussian mixture benchmark datasets, we show in Table I that PacGAN improves significantly over competing methods.

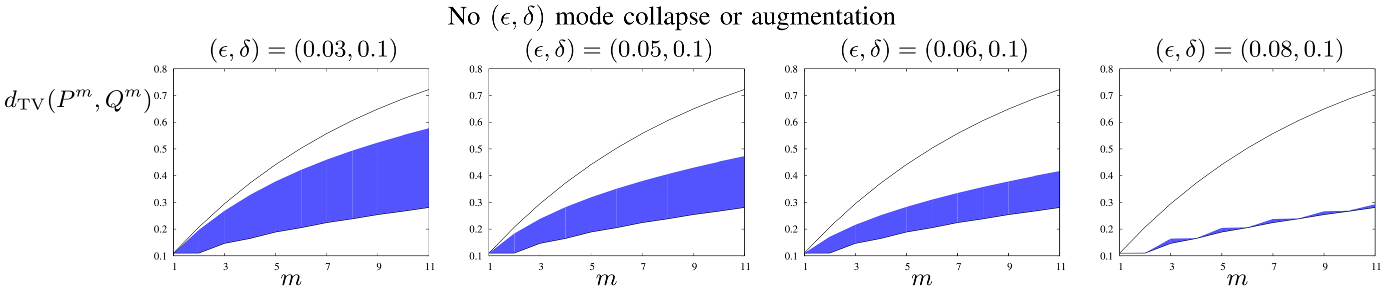

Fig. 6. The evolution of total variation distance over the packing degree m for pairs with no mode collapse/augmentation is shown as a blue band, as defined by the optimization (14) and computed using Theorem 5. For a fixed $d_{\mathrm{TV}}(P, Q) = \tau = 0.11$ and the lack of $(\varepsilon, \delta = 0.1)$-mode collapse/augmentation constraints, we show the evolution with different choices of $\varepsilon \in \{0.03, 0.05, 0.06, 0.08\}$. The black solid lines show the maximum/minimum total variation in the optimization (4) 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.

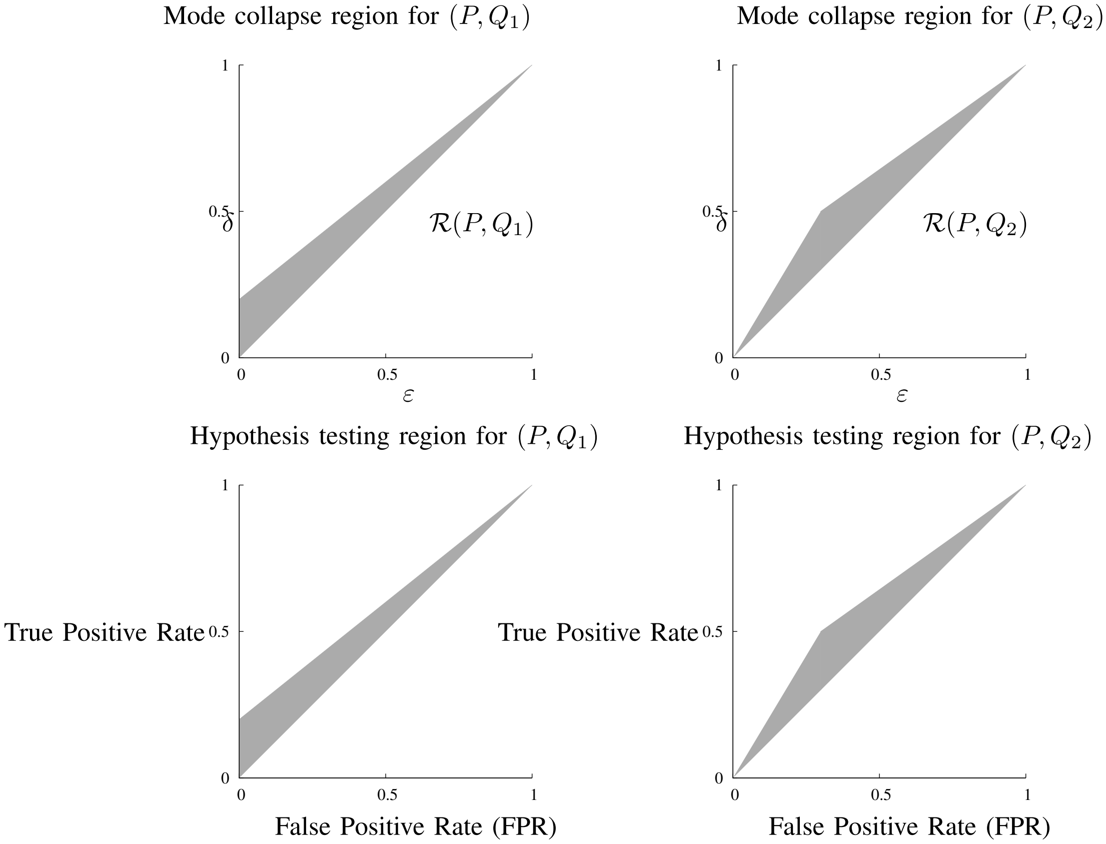

Fig. 7. The hypothesis testing region of (P, Q) (bottom row) is the same as the mode collapse region (top row). The region for mode-collapsed example in Figure 2 (P, Q_1) is shown on the left and non-mode-collapsed example (P, Q_2) on the right.

"Modes" refers to the number of modes captured by the trained generators (higher the better), "high quality samples " refers to the number of samples in the vicinity of the centers of the ground truth mixture of Gaussians (higher the better), and "reverse KL" measures the reverse KL divergence on a quantized version of the generated samples and real data (lower the better).

Another popular benchmark dataset is StackedMNIST, where each (training) sample is constructed by concatenating three randomly chosen MNIST handwritten digits, each one on one of the three color channels: red, green, and blue. PacDCGAN2 generates sharp and diverse images as shown in Figure 9. Further, PacDCGAN significantly improves on both number of modes captures, and the KL divergence as shown in Table II.

V. Discussion

In this work, we propose a packing framework that theoretically and empirically mitigates mode collapse with low overhead. Our analysis leads to several interesting open questions, including how to apply these analysis techniques to more general classes of loss functions such as Jensen-Shannon divergence and Wasserstein distances. Another important question is what packing architecture to use. For instance, a

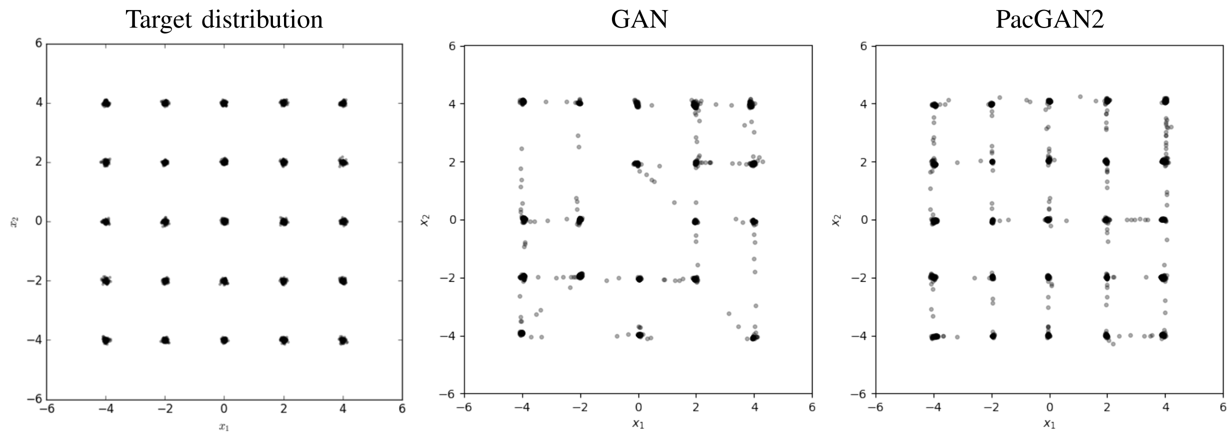

Fig. 8. Scatter plot of the 2D samples from the true distribution (left) of 2D-grid and the learned generators using GAN (middle) and PacGAN2 (right). PacGAN2 captures all of the 25 modes.

	2D-ring			2D-grid		
	Modes (Max 8)	high quality samples	reverse KL	Modes (Max 25)	high quality samples	reverse KL
GAN [2]	6.3±0.5	98.2±0.2 %	0.45±0.09	17.3±0.8	94.8±0.7 %	0.70±0.07
ALI [18]	6.6±0.3	97.6±0.4 %	0.36±0.04	24.1±0.4	95.7±0.6 %	0.14±0.03
Minibatch Disc. [15]	4.3±0.8	36.6±8.8 %	1.93±0.11	23.8±0.5	79.9±3.2 %	0.17±0.03
PacGAN2	7.9±0.1	95.6±2.0 %	0.07±0.03	23.8±0.7	91.3±0.8 %	0.13±0.04
PacGAN3	7.8±0.1	97.7±0.3 %	0.10±0.02	24.6±0.4	94.2±0.4 %	0.06±0.02
PacGAN4	7.8±0.1	95.9±1.4 %	0.07±0.02	24.8±0.2	93.6±0.6 %	0.04±0.01

TABLE I

Two measures of mode collapse proposed in [19] for two synthetic mixtures of Gaussians: number of modes captured by the generator and percentage of high quality samples, as well as reverse KL. Our results are averaged over 10 trials shown with the standard error. We note that 2 trials of MD in 2D-ring dataset cover no mode, which makes reverse KL intractable. This reverse KL entry is averaged over the other 8 trials.

	Stacked MNIST	
	Modes (Max 1000)	KL
DCGAN [35]	99.0	3.40
ALI [18]	16.0	5.40
Unrolled GAN [20]	48.7	4.32
VEEGAN [19]	150.0	2.95
Minibatch Discrimination [15]	24.5±7.67	5.49±0.418
DCGAN (our implementation)	78.9±6.46	4.50±0.127
PacDCGAN2 (ours)	1000.0±0.00	0.06±0.003
PacDCGAN3 (ours)	1000.0±0.00	0.06±0.003
PacDCGAN4 (ours)	1000.0±0.00	0.07±0.005

TABLE II

Two measures of mode collapse proposed in [19] for the stacked MNIST dataset: number of modes captured by the generator and reverse KL divergence over the generated mode distribution. The DCGAN, PacDCGAN, and MD results are averaged over 10 trials, with standard error reported.

framework that provides permutation invariance (e.g., graph neural networks[43], [44], [45] or deep sets [46]) may give better results.

Acknowledgment

This work was also supported in part by by Siemens AG. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575. Specifically, it used the Bridges system, which is supported by NSF award number ACI-1445606, at the Pittsburgh Supercomputing Center (PSC). This work is supported by NSF grants RI-1929955 and CCF-1927712.

References

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

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *NeurIPS*, 2014, pp. 2672–2680.

[3] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv:1312.6114*, 2013.

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

[5] E. L. Denton, S. Chintala, R. Fergus *et al.*, “Deep generative image models using a laplacian pyramid of adversarial networks,” in *NeurIPS*, 2015, pp. 1486–1494.

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

[7] C. Vondrick, H. Pirsiavash, and A. Torralba, “Generating videos with scene dynamics,” in *NeurIPS 29*, 2016, pp. 613–621.

[8] 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:1609.04802*, 2016.

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

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *NeurIPS*, 2013, pp. 3111–3119.

[11] D. P. Kingma and P. Dhariwal, “Glow: Generative flow with invertible 1x1 convolutions,” *arXiv:1807.03039*, 2018.

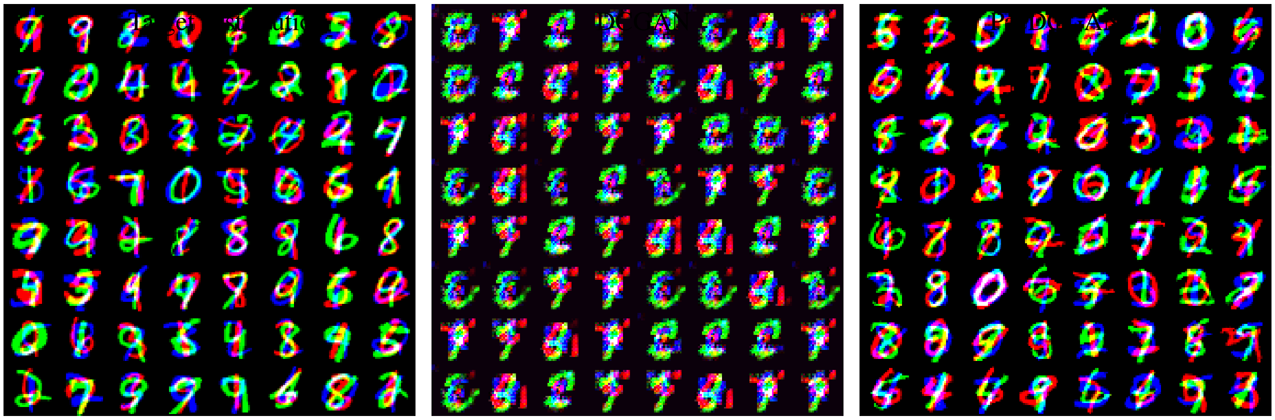

Fig. 9. True distribution (left), DCGAN generated samples (middle), and PacDCGAN2 generated samples (right) from the stacked-MNIST dataset show PacDCGAN2 captures more diversity while producing sharper images.

[12] A. Ilyas, A. Jalal, E. Asteri, C. Daskalakis, and A. G. Dimakis, "The robust manifold defense: Adversarial training using generative models," *arXiv:1712.09196*, 2017.

[13] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *ICCV*, 2015, pp. 3730–3738.

[14] A. Krizhevsky, "Learning multiple layers of features from tiny images," Master's thesis, University of Toronto, 2009.

[15] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *NeurIPS*, 2016, pp. 2234–2242.

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

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

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

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

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

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

[22] Y. Saatci and A. Wilson, "Bayesian gans," in *NeurIPS*, 2017, pp. 3624–3633.

[23] T. Nguyen, T. Le, H. Vu, and D. Phung, "Dual discriminator generative adversarial nets," in *NeurIPS*, 2017, pp. 2667–2677.

[24] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *arXiv:1701.07875*, 2017.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," in *NeurIPS*, 2014, pp. 2879–2887.

[32] ——, "Secure multi-party differential privacy," in *NeurIPS*, 2015.

[33] ——, "The composition theorem for differential privacy," *IEEE Transactions on Information Theory*, vol. 63, no. 6, pp. 4037–4049, June 2017.

[34] Z. Lin, A. Khetan, G. Fanti, and S. Oh, "Pacgan: The power of two samples in generative adversarial networks," *arXiv:1712.0408*, 2017.

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

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

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

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

[39] K. Mills and I. Tamblyn, "Phase space sampling and operator confidence with generative adversarial networks," *arXiv:1710.08053*, 2017.

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

[41] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed sensing using generative models," *arXiv:1703.03208*, 2017.

[42] A. Bora, E. Price, and A. G. Dimakis, "Ambientgan: Generative models from lossy measurements," in *ICLR*, 2018.

[43] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NeurIPS*, 2016, pp. 3844–3852.

[44] K. K. Thekumparampil, C. Wang, S. Oh, and L.-J. Li, "Attention-based graph neural network for semi-supervised learning," *arXiv:1803.03735*, 2018.

[45] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv:1609.02907*, 2016.

[46] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," in *NeurIPS*, 2017, pp. 3391–3401.