Self-Diagnosing GAN: Diagnosing Underrepresented Samples in Generative Adversarial Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Despite remarkable performance in producing realistic samples, Generative Adversarial Networks (GANs) often produce low-quality samples near low-density regions of the data manifold, especially for samples with minor features. Many techniques have been developed to improve the quality of generated samples, either by post-processing generated samples or by pre-processing the empirical data distribution, but at the cost of reduced diversity. To promote diversity in sample generation without degrading the overall quality, we propose a simple yet effective method to diagnose and emphasize underrepresented samples during training of a GAN. The main idea is to use the statistics of the discrepancy between the data distribution and the model distribution at each data instance. Based on the observation that the underrepresented samples have a high average discrepancy or high variability in discrepancy, we propose a method to emphasize those samples during training of a GAN. Our experimental results demonstrate that the proposed method improves GAN performance on various datasets, and it is especially effective in improving the quality and diversity of generated samples with minor features.

1 Introduction

Generative Adversarial Networks (GANs) have achieved remarkable performance in producing realistic samples for complex generation tasks, including image/video synthesis [5, 22], style transfer [46, 14], and data augmentation [30]. However, GANs often fail to learn sparse regions of data manifold [16, 9], e.g., the regions composed of samples with minor features, and when sampling from a trained GAN, the quality of samples may vary much. In particular, the minor data subgroups suffer from poor modeling or experience mode collapse more often than the majority group does [44].

Many of previous techniques have focused on improving the overall sample quality of GANs, either by pre-processing the training dataset or by post-processing generated samples. The pre-processing aims to remove instances that cannot be well-represented by GANs even before the training starts and gains fidelity on the focused samples. [9]. A similar idea has been used to truncate the latent space by resampling or moving samples that fall outside of some acceptable range during training [16, 6]. Post-processing, on the other hand, is a technique that can be applied after the training to remove low-quality generated samples by rejection sampling [3, 38]. All these approaches are effective in increasing the overall fidelity of samples from GANs, but reducing the diversity as a trade-off, and may exacerbate biases against the minor subgroups in sample generation.

In this work, we aim to improve diversity in sample generation without degrading the overall quality, with a special focus on coverage and quality improvement for minor features. Toward this, we design methods to detect and emphasize underrepresented samples in training of GANs. Due to the latent nature of features and the lack of explicit labels available, detecting minor subgroups is especially challenging for unsupervised learning. Therefore, we first develop two new metrics, which

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

can be easily calculated from a discriminator output of GANs, to detect underrepresented minor data subgroups. The main idea is to measure the statistics (mean and variance) of the estimated discrepancy between the data distribution and model distribution at each data instance over multiple epochs of the training. The mean discrepancy indicates how close the data distribution is to the model distribution at each data over the training, while the variance in discrepancy measures how such discrepancy fluctuates across the training. We provide theoretical and empirical evidence that the mean discrepancy can effectively detect underrepresented samples, especially near collapsed modes, while the variance in discrepancy can detect minor data instances, which GANs suffer from modeling.

Based on these observations, we propose a novel method to emphasize underrepresented samples during the training of GANs by score-based weighted sampling, where the score is defined as a weighted sum of the two metrics we devised. We validate our method with thorough experiments over controlled and real datasets and demonstrate the efficacy of the proposed sampling method in improving not only the overall quality (both fidelity and diversity combined) of sample generation but also the coverage and quality for minor features. Our contributions can be summarized as follows.

- We propose two new metrics, which can be simply computed from the discriminator, to diagnose GAN training and to detect underrepresented samples. By theoretical analysis and controlled experiments, we demonstrate that the proposed metrics are effective in detecting underrepresented minor samples.
- We propose an algorithm that can effectively emphasize underrepresented data by score-based weighted sampling during the training of GANs. Our experiments on controlled and real datasets show that our method improves diverse performance metrics on several GAN variants and enhances the coverage and quality of minor feature generation.

2 Related Work

Promoting data coverage in GANs Due to the unstable nature of the min-max game between a generator and a discriminator, GANs often suffer from mode collapse and produce samples with poor diversity. Several approaches have been proposed to promote better data coverage by modifying architectures [20, 23], loss functions [2, 1] or adding regularizations [8, 4, 36]. While effective in promoting overall data coverage, these approaches do not provide special care on minor modes and often fail to recover them when the minority ratio for certain feature is extremely low. We provide a method to promote data coverage for minor features even when the minority ratio is significantly low.

There exists another line of works to improve data coverage by designing hybrid generative models [33, 28, 44], which combines the idea of reconstructive models (e.g. variational autoencoder) to GANs, to take advantages of the reconstructive models in recovering diverse modes. This hybrid method, however, requires relatively high computational overhead to guarantee data coverage for all (or partial) real modes by optimizing reconstruction error in feature domain. Our method directly detects and emphasizes underrepresented samples so that the computational overhead is much lower.

Improving GAN performance by diagnosing samples There have been promising attempts to improve GAN training by using the discriminator outputs to estimate the discrepancy between the data distribution and implicit model distribution. DRS [3] proposes the density ratio estimate based on the discriminator output to apply rejection sampling to filter generated samples. GOLD [26] uses the similar estimate to re-weight fake samples to emphasize underrepresented fake samples. In [10] and [12], on the other hand, an external classifier is used to improve the density ratio estimates. There also exist some approaches to use discriminator outputs to select or weight "useful" fake samples during training. Top-k training [31] updates the generator by using only top-k fake samples with the largest discriminator outputs. In [32] and [41], discriminator-based importance re-weighting schemes for fake samples are developed, and in [40], latent samples are optimized to improve the fidelity.

Our method uses the discrepancy estimate proposed in [3], but its empirical mean and variance over multiple epochs, to extract more reliable and useful information to detect underrepresented minor data subgroups. We provide theoretical evidence of why not only the mean but also the variance of discrepancy estimate is effective in detecting underrepresented samples. Our method detects and emphasizes underrepresented real samples, not the fake samples. This difference is significant in promoting the data coverage of minor subgroups, since when fake samples already fail to cover minor modes, emphasizing a subset of fake samples cannot improve the data coverage for missed modes.

3 Two Metrics to Detect Underrepresented Samples During GAN Training

3.1 Measuring the discrepancy of GANs

GAN training aims to train a generator with an implicit model distribution $p_g(x)$ that closely matches the data distribution $p_{\mathsf{data}}(x)$. The discrepancy between $p_{\mathsf{data}}(x)$ and $p_g(x)$ can be measured by the log density ratio $\log(p_{\mathsf{data}}(x)/p_g(x))$, but it cannot be directly calculated in GANs, since $p_{\mathsf{data}}(x)$ is unknown and $p_g(x)$ is implicit. Instead, the analysis in the original GAN paper [11] can be used to define an estimate on the density ratio by using the discriminator output as explained in [3].

The original GAN solves the min-max optimization $\min_G \max_D V(D,G)$ for the loss $V(D,G) = \mathbb{E}_{x\sim p_{\mathsf{data}}}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1-D(G(z)))]$. For any fixed generator G, the optimal discriminator yields $D^*(x) = \frac{p_{\mathsf{data}}(x)}{p_{\mathsf{data}}(x)+p_g(x)}$ and this allows us to define the Log-Density-Ratio estimate (LDR) by

$$\mathsf{LDR}(x) := \log(D(x)/(1-D(x))). \tag{1}$$

When $D(x) = D^*(x)$, the $\mathsf{LDR}(x)$ is equal to the log density ratio $\log(p_{\mathsf{data}}(x)/p_g(x))$. When $\mathsf{LDR}(x) > 0$, the data point x is underrepresented in the model, i.e., $p_{\mathsf{data}}(x) > p_g(x)$, while when $\mathsf{LDR}(x) < 0$, the data is overrepresented, i.e., $p_{\mathsf{data}}(x) < p_g(x)$. Thus, we can leverage the value of $\mathsf{LDR}(x)$ of each instance x to give feedback to improve the generator if the estimation is valid.

Some prior works have used the LDR estimate to improve GAN training. As an example, GOLD [26] uses $\mathsf{LDR}(x)$ to evaluate the quality of the fake samples and re-weights the underestimated fake samples when training the generator for conditional GANs. However, we later show that re-weighting fake samples is less effective than re-weighting real samples in improving diversity in sample generation. We also empirically show that $\mathsf{LDR}(x)$ is an unstable metric to use. More detailed arguments are available in the Appendix §A.

As a remedy, we propose to use statistics of $\mathsf{LDR}(x)$, which are much more stable and informative metrics, to detect underrepresented data regions during the training. The main intuition is to use training dynamics–the behavior of a model as training progresses–to diagnose the learning behavior of each sample. In supervised learning, training dynamics have been widely studied to detect "hard-to-learn" samples [7, 34, 39]. However, in learning generative models, the metrics to diagnose training dynamics are not clear since there is no explicit reference to measure the accuracy of the model. Here we define metrics that estimate the mean and variance of the discrepancy of GANs, LDRM (LDR Mean) and LDRV (LDR Variance), at each sample x across the training steps $T = \{t_s, ..., t_e\}$:

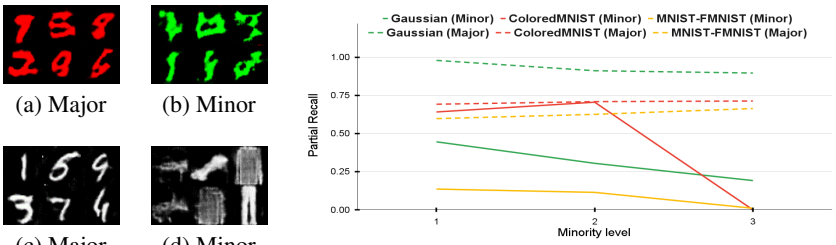
$$\mathsf{LDRM}(x;T) = \frac{1}{|T|}\sum_{k\in T}\mathsf{LDR}(x)_k, \quad \mathsf{LDRV}(x;T) = \frac{1}{|T|-1}\sum_{k\in T}\left[\mathsf{LDR}(x)_k - \mathsf{LDRM}(x;T)\right]^2, \tag{2}$$

where $\mathsf{LDR}(x)_k$ is the recorded LDR estimate (1) in the k-th training step. $\mathsf{LDRM}(x)$ measures how close $p_{\mathsf{data}}(x)$ is to $p_g(x)$ over the training at sample point x, while $\mathsf{LDRV}(x)$ measures how such discrepancy fluctuates across training.

Intuitively, samples that have been well-learned and generalized will have consistently small $\mathsf{LDR}(x)$ since $D(x) \approx 1/2$ (i.e., $p_{\mathsf{data}}(x) \approx p_g(x)$), thus will exhibit low LDRM and LDRV, while underrepresented "hard-to-learn" samples will show high LDRM or LDRV values. In the rest of this section, we thoroughly study the characteristics of data instances with high LDRM or high LDRV.

3.2 LDRV is effective in detecting minor features

GANs have poor modeling for minor samples GANs are known to struggle with modeling minor samples [16]. To scrutinize this phenomenon, we use datasets for which we can control the minority factor. In specific, we use following datasets: (1) Single-mode Gaussian, (2) A mixture of MNIST (major) and FMNIST (minor), and (3) Colored MNIST with Red (major) and Green (minor) samples. Each dataset has major and minor groups divided with certain factor. Details of dataset setup are available in the Appendix §F. Figure 1 shows the poor quality of generated samples with minor features, relative to major features. To quantify the level of underrepresentedness, we examine the coverage of modes for major vs. minor groups with the Partial Recall [18], which is the portion of the subset of real samples that reside in the manifold of the fake samples. As shown in Figure 1e, major and minor groups have large recall gap and the gap gets worsen as the minority level gets severe. This observation indicates that the minor group suffers not only the poor quality problem but also the low coverage problem, and it gives a strong motivation to detect the minor samples and emphasize them.



(a) Major (b) Minor (c) Major (d) Minor (e) Partial Recall of major/minor groups vs. minority level

Figure 1: Analysis on generated samples of DCGAN [27] trained with (1) Single-mode Gaussian (2) A mixture of MNIST (major) and FMNIST (minor), and (3) Colored MNIST with Red (major) and Green (minor) samples. (a) $\sim$ (d) show the sample examples with major/minor features, (e) shows the Partial Recall of major (dotted)/minor (solid) samples in each dataset on various minority levels.

LDRV and minor samples We next provide heuristic arguments that LDRV can be used to detect samples with minor features. In particular, we show that minor samples tend to have higher LDRV values. First, we view the discriminator as the logistic regression model: for each input x_i the discriminator takes the inner product between the feature vector $\phi_i = F(x_i)$ and the weight vector θ of the last layer to produce the reality score (the probability that the sample x_i is real ($y_i = 1$)), i.e.,

$$D(x_i;\theta) = 1/(1+e^{-\theta^T\phi_i}) = p(y_i = 1|\phi_i,\theta). \tag{3}$$

From a Bayesian perspective, assuming that prior distribution of θ is $p(\theta) = \mathcal{N}(\theta|0, s_0 I)$, the posterior distribution over θ is given by $p(\theta|(\phi_i, y_i)_{i=1}^n) \propto p(\theta)p(y_1^n|\phi_1^n,\theta)$. To obtain a Gaussian approximation to the posterior distribution, we first find the maximum a posteriori estimate θ_{MAP} that maximizes $\log p(\theta|(\phi_i, y_i)_{i=1}^n)$, which defines the mean of Gaussian. The covariance is then given by the inverse of the matrix of second derivatives of the negative log likelihood, which takes the form

$$S_n = \left(\sum_{i=1}^n D(x_i;\theta)(1-D(x_i;\theta))\phi_i\phi_i^T + \frac{1}{s_0}I\right)^{-1}. \tag{4}$$

Lastly, approximating $D(x_i;\theta)$ in (3) by the Taylor expansion at $\theta = \theta_{\mathsf{MAP}}$, LDRV can be expressed as

$$\mathsf{LDRV}(x_i) \approx \mathrm{var}\left(\log(D(x_i;\theta)/(1-D(x_i;\theta)))\right) \approx \phi_i^T S_n \phi_i. \tag{5}$$

Details of the analysis is available in the Appendix §B.

This analysis shows an important aspect regarding LDRV and minor features. First, (5) shows that as the feature vector ϕ_i becomes more correlated with the principal components of S_n (eigenvectors with largest eigenvalues), its LDRV gets larger. Since each eigenvalue of S_n is the reciprocal of that of S_n^{-1}, we consider the characteristics of the eigenvector v of S_n^{-1} with the least eigenvalue, which is the minimizer of

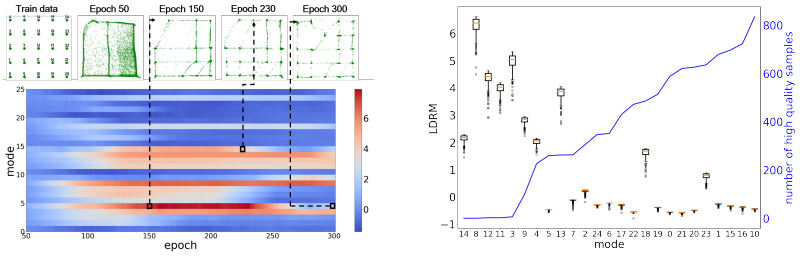
$$\langle y, S_n^{-1} y\rangle = \sum_{i=1}^n D(x_i;\theta)(1-D(x_i;\theta))\langle y, \phi_i\rangle^2 + \text{const.} \tag{6}$$

Eq. (6) shows if y does not align with (or orthogonal to) majority of feature vectors $\{\phi_i\}$ having $D(1-D) > 0$, then it tends to have a smaller eigenvalue. Since a minor feature vector ϕ_j may have a small component on the eigenspace formed by the majority of $\{\phi_i\}$ having $D(1-D) > 0$, when we plug in $y = \phi_j$ into (6), the summation becomes small. This shows that the minor feature vector ϕ_j is correlated with the least eigenvector v of S_n^{-1} and thus it will have higher LDRV.

In Table 1, we show that minor group indeed has higher LDRV. Thus, both theoretical and empirical evidence shows that we can detect minor samples by investigating LDRV of training samples.

Table 1: Averaged LDRV of major/minor groups on various datasets with majority rate 90%.

Group	Gaussian (σ=3.0)	Colored MNIST	MNIST-FMNIST
Major	0.001	0.077	0.082
Minor	0.098	0.186	0.115



(a) LDRM and generated samples

(b) LDRM distribution and the number of high-quality samples over modes

Figure 2: (a) Training dynamics of 25 Gaussians. The index of each mode is equal to $5x + y$ where the coordinates $(x, y) \in \{0, 1, 2, 3, 4\}^2$. LDRM of training samples is recorded for each mode with window $|T| = 50$. Modes with high LDRM do not appear in the generated samples. (b) The empirical distribution of LDRM (box plot) for each mode from training samples, compared with the number of high-quality generated samples from each mode (blue). LDRM can effectively detect dropped modes.

3.3 LDRM is effective in detecting missing modes

Mixture of 25 Gaussians From the definition of LDRM (2), high LDRM samples x tend to have smaller $p_g(x)$ than $p_{\mathsf{data}}(x)$ over the training, thus are underrepresented. We next investigate the ability of LDRM to detect the regions of data manifold not yet covered by the model distribution $p_g(x)$. We consider a mixture of 25 2D isotropic Gaussian distributions [20, 43, 38, 3]. During training, we record $\mathsf{LDR}(x)$ of the training samples and calculate LDRM values with window size $|T| = 50$. We inspect LDRM values averaged over samples of each mode during the training. As shown in Fig. 2a, we observe that samples from underrepresented modes have higher mean LDRM values. This implies that we can detect the mode recovery by inspecting the mean of LDRM values.

To further examine the mode recovery in generated samples, we assign each generated sample to its closest mode and consider it as a "high-quality" sample if it is within four standard deviations from its assigned mode [43, 3]. We then count the number of high-quality samples of each mode among 10,000 generated samples and analyze the correlation between the high-quality sample counts and the distribution of LDRM. As shown in Fig. 2b, modes with only a few high-quality samples tend to have higher LDRM. This indicates that LDRM of the data instances can be used to detect the regions of data manifold not yet covered by the model, even without looking at the generated samples.

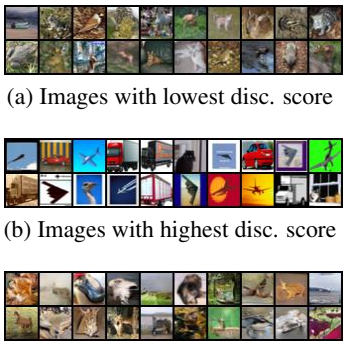
4 Algorithm to Emphasize Underrepresented Samples

4.1 Proposed method: Stochastic Gradient Descent (SGD) sampled by discrepancy

We propose a simple modification to the GAN training procedures by using score-based weighted sampling for mini-batch SGD to emphasize underrepresented samples. Let $\mathcal{D} = \{x_i\}$ be the training dataset. The mini-batch of size B for the training dataset is formed by $\mathcal{D}_B = \{x^{(j)} : x^{(j)} = x_i \text{ where } i \sim P_s(i) \text{ for } j = 1, \ldots, B\}$, i.e., each sample $x_i \in \mathcal{D}$ is sampled with certain probability $P_s(i)$. Our objective is to design the sampling frequency $P_s(i)$ that can emphasize underrepresented samples. Based on the observations in Section 3, we first devise the discrepancy score $s(x_i; T)$ that reflects the underrepresentedness of each sample as follows:

$$s(x_i; T) = \mathsf{LDRM}(x_i; T) + k\sqrt{\mathsf{LDRV}(x_i; T)}, \tag{7}$$

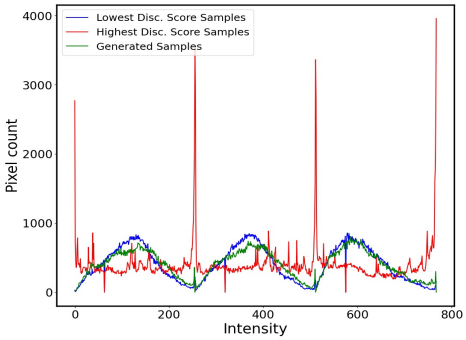
where T is the set of steps used to calculate the discrepancy scores and k is the hyperparameter to modulate the contribution of each statistic. The score (7) can be interpreted as an upper limit of the confidence interval of LDR estimate, or weighted sum of LDRM and the square root of LDRV with weight controlled by k. To ensure every data is sampled with at least some chance, we clip the minimum value of $s(x_i)$ to be $\epsilon = 0.01$ (`min_clip`) and clip the maximum value to have max-min ratio of 50, i.e., $\max s(x)/\min s(x) = 50$ (`max_clip`). For the clipped score $s'(x_i; T) = \texttt{max_clip}(\texttt{min_clip}(s(x_i; T)))$, our final weighted sampling frequency is $P_s(i) = \frac{s'(x_i;T)}{\sum_{j=1}^{|\mathcal{D}|} s'(x_j;T)}$.



(a) Images with lowest disc. score

(b) Images with highest disc. score

(c) Generated samples



(d) Historgram of pixel count over intensity level

Figure 3: Training images with (a) lowest, (b) highest discrepancy scores, and (c) generated samples. Generated samples resemble training images with lowest score. (d) A smoothed histogram of the intensities for 100 samples per group. The intensity levels of RGB channels are concatenated, resulting in total $768 = 256 \times 3$ levels. Images with the lowest scores (blue) and generated samples (green) have a similar distribution, while images with the highest scores (red) show a high discrepancy.

4.2 Sample analysis of the discrepancy score

To check whether our discrepancy score indeed captures the underrepresented samples, we analyze the samples with lowest/highest discrepancy scores. We train SNGAN [25] on CIFAR-10 [17] for 40k steps and measure the discrepancy score of each sample. We first present the images with lowest (Fig. 3a)/highest (Fig. 3b) discrepancy scores among training images, and compare them with generated samples (Fig. 3c). High-scoring images have properties that are distinct from the generated samples (e.g., unusual background or shape), while low-scoring images contain features that are also available in generated samples. Comparing the pixel intensity histogram (Fig. 3d) reveals the difference more clearly in sample properties. Images with lowest discrepancy scores exhibit similar intensity distribution with generated samples, while images with highest scores appear to show an extremely different tendency. We also analyze the Partial FID (FID [13] measured with a subset of training samples) of lowest/highest-score groups. The highest-score group has a Partial FID of 94.63 while the lowest-score group has 22.43. The large gap between the two groups states that the generator fails to generate samples similar to high-score group. These results imply that our discrepancy score successfully identifies underrepresented data that may need emphasis in further training. For more examples of images with scores both for CIFAR-10 and CelebA, see the Appendix §C.

4.3 Post-processing by discriminator rejection sampling with auxiliary discriminator

Our weighted sampling gives bias toward underrepresented samples during training. Though effective in improving diversity, this results in modified data distribution $p'_{\mathsf{data}}(x) = f(x)p_{\mathsf{data}}(x)$ where $f(\cdot)$ is the normalized sampling frequency. Thus, the trained model distribution p_g may be different from the original data distribution p_{data}. To solve this, we utilize the Discriminator Rejection Sampling (DRS) [3] to correct the bias after training. The rejection sampling accepts a generated sample with probability $\frac{p_{\mathsf{data}}(x)}{Mp_g(x)}$ for some constant $M > 0$. To conduct rejection sampling, DRS method needs an estimate for $p_{\mathsf{data}}(x)/p_g(x)$ calculated based on the discriminator outputs. Since our discriminator is trained with biased $p'_{\mathsf{data}}(x)$, we add an auxiliary discriminator and train it with uniform sampling (i.e., without applying our sampling technique) during the weighted sampling procedure to obtain the LDR estimate (1) for DRS, and use this measure for the rejection sampling of generated samples.

4.4 Self-Diagnosing GAN (Dia-GAN)

The overall algorithm (with details in the Appendix §D) can be summarized as below:
Phase 1 - Train and Diagnose: Train GAN and evaluate the discrepancy score for each data instance.
Phase 2 - Score-Based Weighted Sampling: Encourage GAN to learn underrepresented regions of data manifold through score-based weighted sampling (Section 4.1).
Phase 3 - DRS: After GAN training, correct the model distribution $p_g(x)$ by rejection sampling.

Table 2: Comparison of diverse sampling/weighting methods for CIFAR-10/CelebA image generation.

Dataset	CIFAR-10				CelebA					
Methods	SNGAN		SSGAN		SNGAN			SSGAN		
	FID ↓	IS ↑	FID ↓	IS ↑	FID ↓	P ↑	R ↑	FID ↓	P ↑	R ↑
Vanilla	26.90	7.36	22.01	7.65	7.12	0.68	0.44	7.19	0.68	0.44
DRS [3]	24.54	7.57	20.51	7.77	7.04	0.68	0.44	7.08	0.68	0.45
GOLD [26]	28.86	7.21	21.90	7.57	7.31	**0.69**	0.44	7.46	0.68	0.43
GOLD + DRS	24.65	7.53	19.36	7.79	6.97	0.68	0.44	7.15	0.67	0.45
Top-k [31]	24.45	7.60	20.01	7.78	7.35	0.67	0.44	7.23	0.67	0.45
Top-k + DRS	23.92	7.70	20.09	7.88	7.35	0.68	0.44	7.16	**0.68**	0.45
Dia-GAN	**19.66**	**7.95**	**16.31**	**8.14**	**6.70**	0.64	**0.48**	**6.88**	0.66	**0.46**

5 Experiments

5.1 Evaluation metrics and baselines

Evaluation metrics To evaluate the effect of our method on learned model distribution, we use various evaluation metrics including (1) Fréchet Inception Distance (FID) [13], (2) Inception Score (IS) [29], and (3) Precision and Recall (P&R) [18]. In addition to these global evaluation metrics, we consider (4) Reconstruction Error (RE). RE score is calculated by first training a convolutional autoencoder (CAE) with generated samples, and then calculating Euclidean distance between each training data and its reconstruction. RE can assess whether $p_g(x)$ covers $p_{\mathsf{data}}(x)$ since CAE is known to have high RE for out-of-distribution samples [42, 45]. For more details, see the Appendix §E.

Baselines We compare the effect of our method with other methods that use the discriminator output for improving GAN training; 1) DRS [3], 2) Gap of log-densities (GOLD) [26], and 3) Top-k training [31]. GOLD[1] uses the LDR estimate on generated samples to re-weight underrepresented samples (having high LDR) during training of GANs. Top-k training uses only top-k fake samples with the largest discriminator outputs, i.e., the samples believed to be the "most realistic", during the training of the generator. As our algorithm uses DRS after the training, we also analyze each method's performance with post-processing by DRS to measure the exact gain from our sampling method.

5.2 GAN performance enhancement on real datasets

Experiments on CIFAR-10 and CelebA We first assess our method on two widely-studied GAN benchmark datasets, CIFAR-10 [17] and CelebA [21]. We evaluate our method on state-of-the-art GANs; SNGAN [25] and SSGAN [37] with non-saturating variant of the original loss. We train our model for 50k (75k) steps for CIFAR-10 (CelebA), where for our method and GOLD, the phase 1 takes 40k (60k) steps, and the phase 2 takes the remaining. We record LDR every 100 steps and use the last 50 records for calculating the discrepancy score. For the discrepancy score (7), we use $k = 0.3\ (5.0)$ for CIFAR-10 (CelebA). Detailed configurations and hyperparameter search procedure are available in the Appendix §F.

In Table 2, we first compare FID and IS over various methods on the CIFAR-10 dataset. Our proposed Dia-GAN achieves the best FID and IS with a great margin among all baseline methods in every GAN variant. This result demonstrates the wide applicability and effectiveness of our method in improving the overall quality (fidelity and diversity combined) of generated samples. Moreover, the comparison between DRS and our method assures that most of the gain indeed comes from our resampling method. Also, we compare FID and P&R over the methods on CelebA. Our method consistently improves FID over baseline GANs. Precision & Recall analysis shows more detailed reasons for the improvement of FID. Our method consistently improves recall (diversity) but with a slight drop in precision (fidelity). As the increase in diversity is dominant, FID, which measures the

[1] As the original GOLD estimator is designed for conditional GANs [24], we consider the unconditional version by removing the conditional discrepancy term.

Table 3: StyleGAN2 on FFHQ 256x256.

	StyleGAN2	**Dia-StyleGAN2**
FID ↓	14.07	**11.89**

Table 4: HingeGAN on CIFAR-10 and CelebA.

	CIFAR-10		CelebA
	FID ↓	IS ↑	FID ↓
HingeGAN	21.99	7.67	6.66
Dia-HingeGAN	**18.74**	**8.02**	**5.98**

combined effect of fidelity and diversity, is consistently improved with our method compared to the baselines. Examples of generated samples from our Dia-GAN are also available in the Appendix §G.

Experiments on StyleGAN2 We further evaluate the scalability of our method with StyleGAN2 [16] on FFHQ 256x256 [15] dataset. We train the model for 250k steps in total where phase 1 takes 200k steps and the phase 2 takes the remaining steps. We set the hyperparameter $k = 3.0$. Our method improves the FID of StyleGAN2 from 14.07 to 11.89 as shown in Table 3. This indicates that our method successfully scales to large state-of-the-art GANs and high resolution images.

Extension to hinge loss We further conduct experiments to show the applicability of our method to other GAN losses. Here, we focus on a commonly used loss, the hinge loss (HingeGAN) [19, 35]. Our method is not directly applicable to the hinge loss since the output of the optimal discriminator $D_h(x)$ is not $\frac{p_{\mathsf{data}}(x)}{p_{\mathsf{data}}(x)+p_g(x)}$ anymore. Instead, $D_h(x)$ is 1 if $p_{\mathsf{data}}(x) > p_g(x)$ and -1 if $p_{\mathsf{data}}(x) < p_g(x)$. One possible workaround is attaching an auxiliary layer to the discriminator and training it with the original GAN loss. However, we instead present empirical evidence showing that $D_h(x)$ itself still contains useful information about the degree of learning for the input x. We consider the variant of our method, Dia-HingeGAN, by calculating the mean and variance of $D_h(x)$ and using the same scoring rule of (7). In Table 4, we compare the performance of HingeGAN and Dia-HingeGAN with the same configuration of the previous experiment. Interestingly, our method shows significant improvement in both CIFAR-10 and CelebA. This implies that despite the optimal form of the discriminator is different, the statistics of its output still provide meaningful information about the underrepresented features. We leave the theoretical analysis of this variant method as a future work.

5.3 Minor feature generation

Controlled experiments As our method emphasizes underrepresented samples in GAN training, we evaluate how much our method helps the generation of minor samples. To control the level of minority, we design a Colored MNIST dataset with red (major) and green (minor) samples, and MNIST-FNIST dataset with MNIST (major) FMNIST (minor) samples, with the majority rates $\rho \in \{90, 95, 99\}\%$. We compare our method with the same set of baseline methods as in Section 5.1. Additionally, we compare our method with PacGAN [20], the approach to handle the mode collapse problem, and with Inclusive GAN [44], which also improves the data coverage over the minority groups by using a hybrid generative model.

Table 5 shows the results of each method in various majority rates. Here, we focus on the reconstruction error (RE) score of minor training samples (green samples for Colored MNIST, and FMINST samples for MNIST-FMNIST dataset). For the training dataset with the majority rate of 99%, our method shows a significant improvement in RE score as only our method succeeds in generating minor samples while others fail. When the majority rate ρ decreases to 95% and 90%, vanilla model starts to generate minor samples but in low quality. For these rates, our method also shows improvement on the quality of generated samples with minor features, resulting in better RE scores. This result implies the efficacy of our method in improving the quality of generated samples with underrepresented features. Examples of samples with minor features for each method are available in the Appendix §H, and detailed configuration of the experiment is available in the Appendix §F.

CelebA minor attribute analysis For the real-world example, we analyze how our method changes the generation of minor attributes of the CelebA [21] dataset, using the meta-information available in the dataset. Specifically, we focus on how much our method improves the occurrence rate of minor attributes, as they usually appear in a much lower rate than its actual ratio. We train a binary classifier

Table 5: Reconstruction Error (RE) score of green training samples in Colored MNIST and FMNIST samples in a mixture of MNIST and FMNIST on different majority rate ρ.

Dataset	Colored MNIST			MNIST-FMNIST		
Majority rate ρ	99%	95%	90%	99%	95%	90%
Vanilla	0.838	0.236	0.218	0.290	0.227	0.215
GOLD [26]	0.813	0.297	0.200	0.296	0.241	0.218
Top-k [31]	0.831	0.210	0.223	0.281	0.232	0.221
PacGAN [20]	0.810	0.244	0.233	0.313	0.251	0.225
Inclusive GAN [44]	0.812	0.274	0.216	0.283	0.230	0.220
Dia-GAN	**0.224**	**0.204**	**0.197**	**0.264**	**0.219**	**0.206**

Table 6: CelebA minor attribute analysis. Averaged LDRV and averaged discrepancy score of CelebA samples with (W/) or without (W/O) minor attributes. O stands for the occurrence of minor attributes among the generated samples in percentage (%) and R stands for the Partial Recall.

	Score				Method			
	LDRV		Discrepancy		Vanilla		**Dia-GAN**	
	W/	W/O	W/	W/O	O ↑	R ↑	O ↑	R ↑
Bald (2.244%)	**0.271**	0.184	**2.938**	2.221	0.678	0.353	**0.836**	**0.393**
Double Chin (4.669%)	**0.219**	0.184	**2.525**	2.224	0.440	0.411	**0.522**	**0.461**
Eyeglasses (6.512%)	**0.254**	0.181	**2.783**	2.200	3.300	0.400	**4.053**	**0.449**
Gray Hair (4.195%)	**0.211**	0.185	**2.450**	2.228	2.273	0.402	**2.369**	**0.436**
Mustache (4.155%)	**0.242**	0.183	**2.699**	2.218	0.157	0.391	**0.228**	**0.433**
Pale Skin (4.295%)	**0.190**	0.186	**2.240**	2.238	0.346	0.380	**0.453**	**0.427**
Wearing Hat (4.846%)	**0.357**	0.177	**3.651**	2.164	2.307	0.380	**3.595**	**0.408**

for each attribute to have train and test accuracy above 95%. We also evaluate the Partial Recall of the minor attributes, since minor samples suffer low-recall problem as explained in Section 3.2.

As shown in Table 6, our method improves both the occurrence (O) and Partial Recall (R) rates of various minor attributes. Moreover, as explained in Section 3.2, minor samples do have higher LDRV (2) and the discrepancy score (7). This indicates that our method indeed captures the underrepresented minor features and successfully promotes the generation of such features during training of GANs. Figure 4 shows examples of generated samples with minor feature appeared by our Dia-GAN.

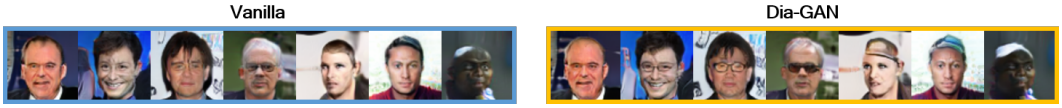


Figure 4: Generated samples of vanilla SNGAN and Dia-SNGAN trained on CelebA.

6 Discussion

We proposed two new metrics, LDRV and LDRM, that can detect underrepresented samples and devised a simple approach to emphasize detected underrepresented samples. Our method successfully improves overall quality of the generated samples in terms of FID and IS, and promotes generation of minor samples. However, we still find the trade-off relationship between the precision and recall of generated samples (Table 2). We leave the investigation of other approaches to use the knowledge of detected underrepresented samples for further improvement of GAN training as a future work.

Our work sheds light on the important but less investigated subject, the underrepresentation of minor samples in training of GANs. We propose a possible approach to solve this problem with feasible computational cost. We think that our work may result in a positive societal impact as incorporating minorities into the consideration of machine learning research is an important objective.

References

[1] Jonas Adler and Sebastian Lunz. Banach wasserstein GAN. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems*, pages 6755–6764, 2018.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223. PMLR, 2017.

[3] Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian Goodfellow, and Augustus Odena. Discriminator rejection sampling. In *International Conference on Learning Representations*, 2019.

[4] David Berthelot, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR*, 2019.

[7] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems*, pages 1002–1012, 2017.

[8] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.

[9] Terrance DeVries, Michal Drozdzal, and Graham W. Taylor. Instance selection for gans. In *Advances in Neural Information Processing Systems*, 2020.

[10] Xin Ding, Z Jane Wang, and William J Welch. Subsampling generative adversarial networks: Density ratio estimation in feature space with softplus loss. *IEEE Transactions on Signal Processing*, 68:1910–1922, 2020.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[12] Aditya Grover, Jiaming Song, Ashish Kapoor, Kenneth Tran, Alekh Agarwal, Eric Horvitz, and Stefano Ermon. Bias correction of learned generative models using likelihood-free importance weighting. In *Advances in Neural Information Processing Systems*, pages 11056–11068, 2019.

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410. Computer Vision Foundation / IEEE, 2019.

[16] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8107–8116. IEEE, 2020.

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[18] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, pages 3929–3938, 2019.

[19] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[20] Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. PacGAN: The power of two samples in generative adversarial networks. *IEEE Journal on Selected Areas in Information Theory*, 1(1):324–335, 2020.

[21] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[22] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[23] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. In *ICLR (Poster)*. OpenReview.net, 2017.

[24] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[25] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *6th International Conference on Learning Representations, ICLR*. OpenReview.net, 2018.

[26] Sangwoo Mo, Chiheon Kim, Sungwoong Kim, Minsu Cho, and Jinwoo Shin. Mining gold samples for conditional gans. In *Advances in Neural Information Processing Systems*, pages 6170–6181, 2019.

[27] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[28] Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, and Shakir Mohamed. Variational approaches for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1706.04987*, 2017.

[29] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[30] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016.

[31] Samarth Sinha, Zhengli Zhao, Anirudh Goyal ALIAS PARTH GOYAL, Colin A Raffel, and Augustus Odena. Top-k training of gans: Improving gan performance by throwing away bad samples. *Advances in Neural Information Processing Systems*, 33, 2020.

[32] Jiaming Song and Stefano Ermon. Bridging the gap between f-gans and wasserstein gans. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9078–9087. PMLR, 2020.

[33] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U. Gutmann, and Charles Sutton. VEEGAN: reducing mode collapse in gans using implicit variational learning. In *NIPS*, pages 3308–3318, 2017.

[34] Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 9275–9293, 2020.

[35] Dustin Tran, Rajesh Ranganath, and David M. Blei. Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems*, pages 5523–5533, 2017.

[36] Ngoc-Trung Tran, Tuan-Anh Bui, and Ngai-Man Cheung. Dist-gan: An improved GAN using distance constraints. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference*, volume 11218 of *Lecture Notes in Computer Science*, pages 387–401. Springer, 2018.

[37] Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Linxiao Yang, and Ngai-Man Cheung. Self-supervised GAN: analysis and improvement with multi-class minimax game. In *Advances in Neural Information Processing Systems*, pages 13232–13243, 2019.

[38] Ryan Turner, Jane Hung, Eric Frank, Yunus Saatchi, and Jason Yosinski. Metropolis-hastings generative adversarial networks. In *International Conference on Machine Learning*, pages 6345–6353. PMLR, 2019.

[39] Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. When do curricula work? *arXiv preprint arXiv:2012.03107*, 2020.

[40] Yan Wu, Jeff Donahue, David Balduzzi, Karen Simonyan, and Timothy Lillicrap. Logan: Latent optimisation for generative adversarial networks. *arXiv preprint arXiv:1912.00953*, 2019.

[41] Yue Wu, Pan Zhou, Andrew Gordon Wilson, Eric P. Xing, and Zhiting Hu. Improving GAN training with probability ratio clipping and sample reweighting. In *Advances in Neural Information Processing Systems*, 2020.

[42] Yan Xia, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *2015 IEEE International Conference on Computer Vision, ICCV*, pages 1511–1519. IEEE Computer Society, 2015.

[43] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

[44] Ning Yu, Ke Li, Peng Zhou, Jitendra Malik, Larry Davis, and Mario Fritz. Inclusive gan: Improving data and minority coverage in generative models. In *European Conference on Computer Vision*, pages 377–393. Springer, 2020.

[45] Chong Zhou and Randy C. Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674. ACM, 2017.

[46] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3.2 and Appendix §B.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section 3.2 and Appendix §B.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] They are contained in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix §F.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Appendix §I.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix §D.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes]
(b) Did you mention the license of the assets? [Yes]
(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]