Collaborative Sampling in Generative Adversarial Networks

Yuejiang Liu,* Parth Kothari,* Alexandre Alahi
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Abstract

The standard practice in Generative Adversarial Networks (GANs) discards the discriminator during sampling. However, this sampling method loses valuable information learned by the discriminator regarding the data distribution. In this work, we propose a collaborative sampling scheme between the generator and the discriminator for improved data generation. Guided by the discriminator, our approach refines the generated samples through gradient-based updates at a particular layer of the generator, shifting the generator distribution closer to the real data distribution. Additionally, we present a practical discriminator shaping method that can smoothen the loss landscape provided by the discriminator for effective sample refinement. Through extensive experiments on synthetic and image datasets, we demonstrate that our proposed method can improve generated samples both quantitatively and qualitatively, offering a new degree of freedom in GAN sampling.

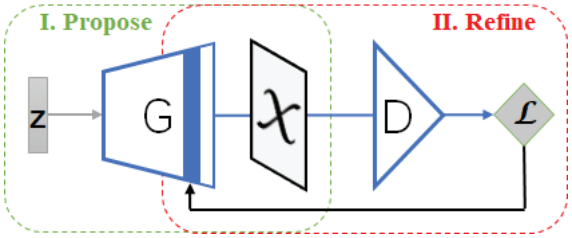


Figure 1: Once training completes, we use both the generator and the discriminator for collaborative sampling. Our scheme consists of one sample proposal step and multiple sample refinement steps. (I) The fixed generator proposes samples. (II) Subsequently, the discriminator provides gradients, with respect to the activation maps of the proposed samples, back to a particular layer of the generator. Gradient-based updates are performed iteratively.

Introduction

Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) are a powerful class of deep generative models known for producing realistic samples. Despite successful applications in a wide variety of tasks (Zhu et al. 2017; Brock, Donahue, and Simonyan 2019; Karras, Laine, and Aila 2019), training GANs is notoriously unstable, often impacting the model distribution. Numerous works have attempted to improve GAN training through loss functions (Arjovsky, Chintala, and Bottou 2017), regularization methods (Miyato et al. 2018), training procedures (Karras et al. 2017) as well as model architectures (Radford, Metz, and Chintala 2015; Zhang et al. 2019). Yet, stabilizing GANs at scale remains an open problem. In this work, we go beyond GAN training and explore methods for effective sampling. Our goal is to improve the model distribution by fully exploiting the value contained in the trained networks during sampling.

A standard practice in GAN sampling is to completely discard the discriminator while using only the generator for sample generation. Recent works propose to post-process the model distribution p_g, implicitly defined by the trained generator, using Monte Carlo techniques such as rejection sampling (Azadi et al. 2019) and Metropolis-Hastings independence sampler (Turner et al. 2019). By rejecting undesired samples based on the output of an optimal discriminator, the accept-reject paradigm is able to recover the real data distribution p_r under certain assumptions. However, these methods have several limitations:

- *exactness*: the assumption that the support of p_g includes the support of p_r is often too strong to hold in practice,
- *efficiency*: the accept-reject procedure suffers from low sample efficiency when p_g is statistically distant from p_r,
- *applicability*: rejection cannot be applied to many scenarios where only one sample is produced, *e.g.*, CycleGAN (Zhu et al. 2017).

Drawing inspiration from Langevin (Roberts and Tweedie 1996) and Hamiltonian Monte Carlo methods (Neal 1996), we address these issues by refining, rather than simply rejecting, the generated samples.

Figure 1 illustrates our proposed collaborative sampling scheme between the generator and the discriminator. Once training completes, we freeze the parameters of the generator and refine the proposed samples using the gradients provided by the discriminator. This gradient-based sample refinement can be performed repeatedly at any layer of the generator, ranging from low-level feature maps to the final output space, until the samples look "realistic" to the discriminator.

The performance of our collaborative sampling scheme is dependent on the loss landscape provided by the discriminator. To further improve the sample refinement process, we propose a practical discriminator shaping method that finetunes the discriminator using the refined samples. This shaping method not only enhances the robustness of the discriminator for classification but also smoothens the learned loss

*Equal contribution

landscape, thereby strengthening the discriminator's ability to guide the sample refinement process.

Our sample refinement method is not mutually exclusive with the accept-reject paradigm. An additional rejection step can be applied subsequent to the refinement process for distribution recovery. To ensure the effectiveness of the rejection step, we propose to diagnose the optimality of the discriminator with the Brier Score (Brier 1950) in contrast to the calibration measure used in (Turner et al. 2019).

Through experiments on a synthetic imbalanced dataset where the standard GAN training is prone to mode collapse, we first show that the previous accept-reject methods may fail due to their strict assumptions, whereas our proposed method achieves superior results on both quality and diversity. We further demonstrate that our method can scale to the image domain effectively and provide consistent performance boost across different models including DCGAN (Radford, Metz, and Chintala 2015), CycleGAN (Zhu et al. 2017) and SAGAN (Zhang et al. 2019). Our proposed method can be applied on top of existing GAN training techniques, offering a new degree of freedom to improve the generated samples. Code is available online[1].

Background

Generative Adversarial Networks

Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) consist of two neural networks, namely the generator G and the discriminator D, trained together. The role of the generator G is to transform a latent vector z sampled from a given distribution p_z to a realistic sample $G(z)$, whereas the discriminator D aims to tell whether a sample comes from the generator distribution p_g or the real data distribution p_r. Training GANs is essentially a minimax game between these two players:

$$\min_G \max_D \mathbb{E}_{x\sim p_r}[\log(D(x))] + \mathbb{E}_{z\sim p_z}[1 - \log(D(G(z))].$$

The solution of this optimization problem, under certain conditions, leads to a generator capable of modelling the data distribution p_r. However, training GANs is notoriously unstable in practice due to the complex dynamics of the minimax game. Our goal is to sidestep the training issues and improve the generated samples during the sampling process.

Monte Carlo Methods

Monte Carlo (MC) methods (Fermi and Richtmyer 1948) are a broad family of algorithms which aim to draw a set of i.i.d. samples from a target distribution $p(x)$. When it is hard to directly sample from $p(x)$, one class of MC algorithms are invented to first sample from another easy-to-sample proposal distribution $q(x)$ and subsequently reject some through an accept-reject scheme. Rejection sampling and Metropolis-Hastings independence sampling (Tierney 1994) are two such instances. Rejection sampling draws random samples from $q(x)$ and accepts them with probability $A(x) = p(x)/Mq(x)$ if there exists an $M < \infty$ such that $p(x) \leq Mq(x)$ for all x. The Metropolis-Hastings independence sampler compares a new sample y with the current one x, accepting y with probability $A(x, y) = \min\{1, p(x)q(y)/p(y)q(x)\}$. If the support of $q(x)$ includes the support of $p(x)$, these accept-reject methods are guaranteed to converge to the target distribution. However, their efficiency highly depends on the statistical distance between $q(x)$ and $p(x)$.

For the cases where it is practically difficult to find a good proposal $q(x)$, more sophisticated MC methods that leverage informed local moves to explore the important regions of $p(x)$ are preferred for efficiency. Popular algorithms include Langevin (Roberts and Tweedie 1996) and Hamiltonian MC (Neal 1996), which incorporate the information of the target distribution, in the form of $\nabla \log p(x)$, to construct a gradient-based Markov transition $T(x \to y)$. A sufficient condition for an *ergodic* Markov chain to converge to the target distribution $p(x)$ is *reversibility*, also known as the detailed balance condition, $p(x)T(x \to y) = p(y)T(y \to x)$.

GAN Sampling

The standard GAN sampling process (Goodfellow et al. 2014) draws samples from the generator without the involvement of the discriminator. Recently, (Azadi et al. 2019) proposed a rejection sampling scheme which uses the discriminator to filter out samples that are unlikely to be real. In the ideal setting, computing the acceptance probability is made possible by an optimal discriminator $D^*(x)$ as it yields the density ratio between the target $p_r(x)$ and proposal $p_g(x)$:

$$\frac{p_r(x)}{p_g(x)} = \frac{D^*(x)}{1 - D^*(x)} \tag{1}$$

Another recent work (Turner et al. 2019) proposed to replace the rejection sampling by Metropolis-Hastings independence sampling, leveraging the same knowledge about the density ratio to scale better in high dimensions.

Nevertheless, both these methods rely on an accept-reject principle that inevitably sacrifices *sample efficiency*, *i.e.*, a significant number of generated samples are rejected, and *flexibility*, *i.e.*, the accepted samples are restricted to the data manifold learned by the generator. Our work explores a more involved collaboration scheme between the generator and the discriminator, which exploits the gradient of the density ratio provided by the discriminator to modify the generated samples.

Method

In this section, we describe our collaborative sampling method in GANs that uses both the generator and the discriminator to produce samples (at test time). Subsequently, we introduce a discriminator shaping method that smoothens the loss landscape to enhance the effectiveness of our proposed scheme.

Collaborative Sampling

Consider a generator network that inputs a latent code $z \in \mathbb{R}^m$ and produces an output $x \in \mathbb{R}^n$. It typically consists of multiple layers:

$$\begin{aligned} &G(z) = G_L \circ G_{L-1} \circ \cdots \circ G_1(z), \\ &G_l(x_l) = \sigma(\theta_l \cdot x_l) + b_l, \quad l = 1, 2, \ldots, L, \end{aligned} \tag{2}$$

[1] https://github.com/vita-epfl/collaborative-gan-sampling

Algorithm 1 Collaborative Sampling

Input: a frozen generator G, a frozen discriminator D, the layer index for sample refinement l, the maximum number of steps K, the stopping criterion η
Output: a synthetic sample x
Randomly draw a latent code z
$x^0 \leftarrow \text{ProposeSample}(G, z)$
for $k = 0, 1, \ldots, K-1$ **do**
  if $D(x^k) < \eta$ **then**
    $g_l^k \leftarrow \text{GetGradient}(D, x_l^k)$,
    $x_l^{k+1} \leftarrow \text{UpdateActivation}(g_l^k, x_l^k)$, (Eq. 3)
    $x^{k+1} \leftarrow \text{UpdateSample}(G, x_l^{k+1})$, (Eq. 4)
  else
    break
  end if
end for

where G_l is the lth layer of the generator, x_l is the corresponding activation input, σ is a nonlinear activation function, θ_l and b_l are the model parameters. The input to the first layer is $x_1 = z$ and the output of the last layer is $G_L(x_L) = x$. For a randomly drawn sample from the generator distribution, *i.e.*, $x \sim p_g$, the discriminator outputs a real-valued scalar $D(x)$ which indicates the probability of x to be real. When the generator and the discriminator reach an equilibrium, the generated samples are no longer distinguishable from the real samples, *i.e.*, $D^*(x) = \frac{p_r(x)}{p_r(x)+p_g(x)} = 1/2$. However, such a saddle point of the minimax problem is hardly obtained in practice (Arora et al. 2017), indicating room for improvement over the model distribution p_g.

Our goal is to shift p_g towards p_r through sampling without changing the parameters of the generator. Inspired by the gradient-based MC methods using Langevin (Roberts and Tweedie 1996) or Hamiltonian (Neal 1996) dynamics, we leverage the gradient information provided by the discriminator to continuously refine the generated samples through the following iterative updates:

$$x_l^{k+1} = x_l^k - \lambda \nabla_l \mathcal{L}_G(x_l^k), \tag{3}$$

$$x^{k+1} = G_L \circ G_{L-1} \circ \ldots G_l(x_l^{k+1}), \tag{4}$$

where k is the iteration number, λ is the stepsize, l is the index of the generator layer for sample refinement, $\mathcal{L}_G$ is the loss of the generator, *e.g.*, the non-saturating loss advocated in (Goodfellow et al. 2014):

$$\mathcal{L}_G = -\mathbb{E}_{z \sim p_z}[\log D(G(z))] \tag{5}$$

The iterative sample update consists of two parts: in the backward pass, the discriminator provides the generator with gradient feedback to adjust the activation map of the selected layer l (Eq. 3); in the forward pass, the generator reuses part of its parameters to propose an improved sample (Eq. 4). A pseudo code is summarized in Algorithm 1.

Recall that an optimal discriminator outputs the density ratio between $p_r(x)$ and $p_g(x)$. The iterative updates shift samples to the regions in which $p_r(x)/p_g(x)$ is higher.

Algorithm 2 Discriminator Shaping

Input: a frozen generator G, a pre-trained discriminator D, the batch size m
Output: a fine-tuned discriminator $\tilde{D}$
for number of D shaping iterations **do**
  Draw m refined samples $\{x_c^{(1)}, \ldots, x_c^{(m)}\}$ from the collaborative data distribution $p_c(x)$ according to Algorithm 1
  Draw m real samples $\{x_r^{(1)}, \ldots, x_r^{(m)}\}$ from the real data distribution $p_r(x)$
  Shape the discriminator by minimizing the objective function Eq. 6
end for

In other words, samples are encouraged to move to regions where less samples are produced by the generator but more samples are expected in the real data distribution. Our method forms a closed-loop sampling process, allowing both the generator and the discriminator to contribute to sample generation.

Discriminator Shaping

In the ideal scenario where the loss landscape is smooth and monotonic from p_g to p_r, the generated samples can be easily refined towards the real ones according to the gradient feedback. However, this is not always the case in practice for two reasons:

- In the standard GAN training, the objective of the discriminator is solely to distinguish the real and fake samples. This makes the discriminator prone to overfitting to the generator distribution and less robust in unexplored regions.
- When p_g fails to provide good coverage of the target p_r, the density ratio $p_r(x)/p_g(x)$ grows dramatically in the regions where $p_g(x) \to 0$ and $p_r(x) > 0$, giving the discriminator a false sense of x being highly realistic even when $p_r(x)$ is very small.

As a consequence, the discriminator obtained from standard training may misclassify a poorly refined sample as real and fail to suggest further improvements.

To resolve this issue, we devise a practical discriminator shaping method, the goal of which is to strengthen the discriminator such that it is not only accurate in classifying the generated samples but also capable of effectively guiding the sample refinement process. Given the trained generator and discriminator, we fine-tune the discriminator using the refined samples:

$$\mathcal{L}_D = -\mathbb{E}_{x \sim p_r}[\log D(x)] - \mathbb{E}_{x' \sim p_c}[1 - \log D(x')], \tag{6}$$

where x' is a refined sample and p_c is the refined data distribution obtained from our collaborative sampling scheme.

As outlined in Algorithm 2, we conduct the discriminator shaping and collaborative sampling alternatively. This post-training procedure gradually expands the coverage of the model distribution and enforces the discriminator to generalize and better collaborate with the generator for sample

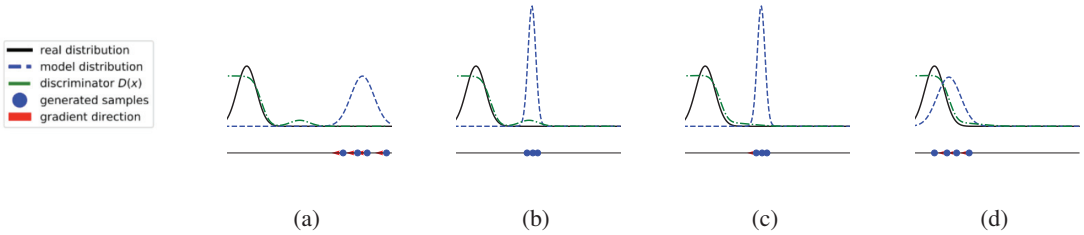


Figure 2: Illustration of our collaborative sampling scheme with discriminator shaping. (a) The trained generator implicitly provides a model distribution that is close to, but not identical to, the real data distribution. At this stage, the gradient provided by the discriminator suggests informed moves. (b) However, the loss landscape from the discriminator may present local optima, hence the sample refinement process ceases. (c) Our discriminator shaping method uses the refined samples to smoothen the loss landscape. (d) The shaped loss landscape is able to better guide the refinement process, shifting the model distribution closer to the target.

refinement. Figure 2 illustrates the proposed method in a simple 1D scenario, showing how the discriminator shaping method using the refined data can help in better approximating the real data distribution.

Discussion

Termination Condition The stopping criterion η in Algorithm 1 can be constructed either deterministically or probabilistically depending upon the objective of the application. In cases where only sample quality matters, *e.g.*, image manipulation, setting η to the median of the discriminator outputs for real samples is a good strategy. On the other hand, when sample diversity is of significant interest, the termination condition can be defined in a probabilistic manner, *e.g.*, stopping the sample refinement process at each step with positive probability. This design choice expands the support of the model distribution $\text{supp}(p_g) \subseteq \text{supp}(p_c)$.

Rejection Step To recover the exact target distribution from the refined model distribution, the acceptance probability of a new refined sample needs to satisfy the detailed balance condition:

$$A(\hat{x}, \hat{y}) = \min\Big\{1, \frac{p_r(\hat{y})}{p_r(\hat{x})} \frac{q(\hat{x}|\hat{y})}{q(\hat{y}|\hat{x})}\Big\}, \tag{7}$$

where $\hat{(\cdot)}$ denotes a refined sample obtained from Algorithm 1, $\hat{x}$ is the currently accepted sample, $\hat{y}$ is the new refined sample, $q(\hat{y}|\hat{x})$ is the transition probability. While the original samples x and y are independently produced by the generator, the refined ones $\hat{x}$ and $\hat{y}$ are no longer independent due to the shared loss function as well as the generator parameters between the respective refinement processes. However, when the state space is sufficiently large, the refinement trajectories $x \to \hat{x}$ and $y \to \hat{y}$ have negligible probability of overlap. Under the assumption that $\hat{x}$ and $\hat{y}$ are independent, we approximate the acceptance probability as in (Turner et al. 2019):

$$\begin{aligned} A(\hat{x}, \hat{y}) &\approx \min\Big\{1, \frac{p_r(\hat{y})}{p_r(\hat{x})} \frac{q(\hat{x})}{q(\hat{y})}\Big\}, \\ &= \min\Big\{1, \frac{D^*(\hat{y})}{D^*(\hat{x})} \frac{1 - D^*(\hat{x})}{1 - D^*(\hat{y})}\Big\}. \end{aligned} \tag{8}$$

Discriminator Diagnosis The accept-reject procedure allows for recovering the target distribution only when the discriminator is optimal. However, it is non-trivial to obtain such a discriminator in practice. Previous work (Turner et al. 2019) proposes to calibrate the trained discriminator and diagnose it with the Z-statistic (Dawid 1997)

$$Z = \frac{\sum_{i=1}^{N} y_i - D(x_i)}{\sqrt{\sum_{i=1}^{N} D(x_i)(1 - D(x_i))}} \tag{9}$$

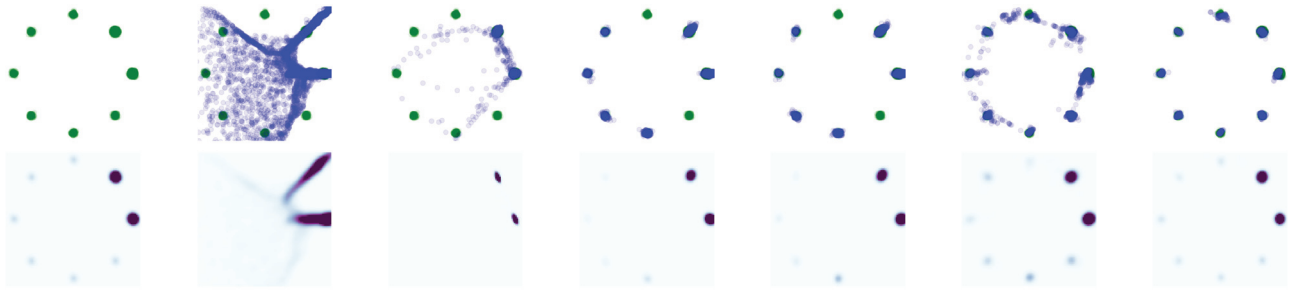
where y_i is the label of sample i, N is the number of samples in the test set.

While having reliable confidence estimates is a necessary condition for the discriminator to reach optimality, it is far from sufficient. One simple counter example is that a random binary classifier is perfectly calibrated on a testset containing an equal amount of real and generated samples, even though it performs poorly in the classification problem. To address this issue, we assess the optimality of the discriminator using the Brier Score (Brier 1950):

$$BS = \frac{1}{N} \sum_{i=1}^{N} (y_i - D(x_i))^2 \tag{10}$$

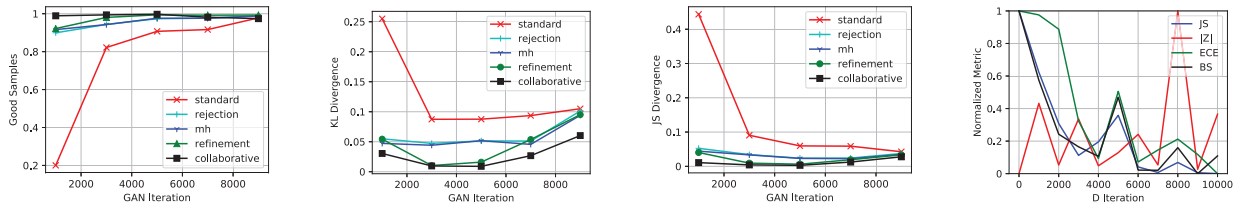
The Brier Score can be decomposed into three terms including not only reliability (calibration) but also resolution and uncertainty (Murphy 1973), thereby measuring the optimality of the discriminator in a broader sense. We employ the Brier Score in the diagnosis of the discriminator before performing the accept-reject step.

Refinement Layer Another key hyperparameter in our method is the index of the generator layer l for sample refinement. On one extreme, we can adjust the proposed sample at the output of the generator, which is equivalent to modifying the sample directly. Manipulating a proposed sample in the data space does not rely on any part of the generator and thus can, in principle, result in an optimal refinement without any constraints. However, shifting a high-dimensional sample from a low-density region to a high-density region in the data space often requires a large number of iterations. On the other extreme, we can choose to



(a) Real (b) GAN-1k (c) GAN-9k (d) Reject-1k (e) MH-1k (f) Refine-1k (g) Collab-1k

Figure 3: Qualitative evaluation of collaborative sampling in GANs on a synthetic imbalanced mixture of eight Gaussians *(green)*. We draw 10k samples *(first row)* from different models and visualize the resulting model distribution using kernel density estimation (KDE) *(second row)*. The output samples *(blue)* from the generator at an early stage of training are not of good quality *(b)*, whereas training GANs longer results in mode collapse *(c)*. Our sample refinement method *(f)* applied to the early terminated GAN not only shifts the proposed samples closer to the real Gaussian components but also expands the categorical coverage. By incorporating the rejection step, *(g)* the full version of our collaborative sampling scheme succeeds in recovering all modes without compromising sample quality, significantly outperforming *(d)* the rejection sampling (Azadi et al. 2019) and *(e)* the Metropolis-Hastings (MH) algorithm with independence sampler (Turner et al. 2019).



(a) Quality (b) Diversity (c) Overall (d) Diagnosis

Figure 4: Quantitative results on the imbalanced mixture of eight Gaussians. Evaluated on 10000 samples. *(a)* Proportion of good samples. Higher is better. *(b)* KL divergence between the categorical distribution of real samples and that of good generated samples. Lower is better. *(c)* JS divergence between the augmented categorical distribution of real samples and that of all generated samples. Lower is better. *(d)* The scores of diagnostic metrics and the performance gain from the rejection step at different stages of the discriminator. We apply the MH method to the generator at 1k and normalize the results on each metric to [0,1] for comparison. Among the three diagnostic metrics, the evolvement of the Brier Score exhibits the strongest similarity to that of the JS divergence.

adjust the latent code z. As the dimension of the latent space is typically much smaller, one can obtain higher computational efficiency. However, this choice restricts the refined samples to the data prior learned by the generator and undermines the assumption of independence between $\hat{x}$ and $\hat{y}$. We empirically find that refining a sample at a middle layer of the generator leads to a good balance between efficiency and flexibility.

Computational Expenses Our collaborative sampling scheme provides higher sample quality at the expense of extra iterations. The additional computational cost not only depends on the choice of the refinement layer and the optimization algorithm but also reflects the quality gap between the proposed samples and refined ones. In the next section, we experimentally show our method can provide considerable improvements within 20 to 50 refinement steps.

Experiments

In this section, we present experimental results to validate the proposed collaborative sampling scheme. We first show that our method outperforms the existing sampling methods on several GAN variants for modeling data distributions. Moreover, we demonstrate the benefits of our method in an image manipulation task, for which the previous accept-reject samplers are not applicable. Finally, we examine the effect of discriminator shaping as well as the choice of the refinement layer.

Synthetic Data

We first evaluate our collaborative sampling scheme on a synthetic 2D dataset, which comprises of an imbalanced mixture of 8 Gaussians. 90% of the real samples are drawn from two Gaussian components while the rest 10% are drawn from the other six. We use a standard fully-connected MLP with 6 hidden layers and 64 hidden units per layer to model the generator and the discriminator. We shape the dis-

Metric	Δ Good		Δ KL		Δ JS	
	Pearson	Spearman	Pearson	Spearman	Pearson	Spearman
Z (Dawid 1997)	0.02 (0.72)	0.11 (0.52)	0.15 (0.30)	-0.01 (0.37)	0.07 (0.41)	-0.06 (0.35)
ECE (Naeini 2015)	-0.18 (0.54)	-0.16 (0.55)	0.32 (0.25)	0.20 (0.42)	0.31 (0.35)	0.24 (0.47)
BS (Brier 1950)	**-0.68 (0.01)**	**-0.72 (0.01)**	**0.51 (0.04)**	0.46 (0.12)	**0.77 (0.00)**	**0.77 (0.00)**

Table 1: Statistical correlation between the performance gain from the rejection step and the score of different diagnostic metrics. We compute the Pearson's and Spearman's correlation coefficients based on the results of the MH method using discriminators fine-tuned for different iterations. Among the three diagnostic metrics, only the Brier Score has significant correlation with the performance gain ($p \leq 0.05$).

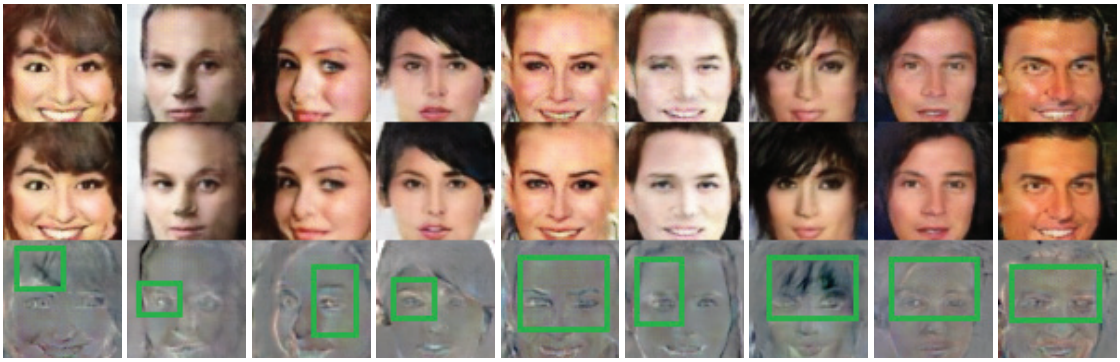


Figure 5: Qualitative results of our collaborative sampling method for DCGAN on the CelebA at 64×64 resolution. The DCGAN model is first trained for 30 epochs. The discriminator is further shaped for one epoch. The generated samples are refined at the 2nd layer of the generator for 50 steps. *(Top)* Samples produced by the generator. *(Middle)* Samples produced by our collaboratively sampling method. *(Bottom)* The differences between the generated and refined images are highlighted for visualization.

criminator for $5k$ additional iterations after terminating the standard GAN training and conduct a maximum 50 sample refinement steps in the data space with a step size of 0.1. For fair comparison, we set the hyperparameter γ in the rejection sampling method (Azadi et al. 2019) to 1.0 and the MC iteration number k in the Metropolis-Hasting method (Turner et al. 2019) to 20. In addition, for fairness with regards to our discriminator shaping, we train the discriminator for $5k$ additional iterations before running these accept-reject methods.

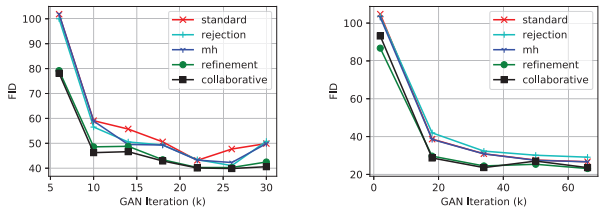
Figure 3 shows the qualitative results of different sampling methods. The standard GAN training gradually runs into mode collapse on the imbalanced dataset, resulting in high sample quality but low diversity after $9k$ iterations. On the other hand, if the training procedure is early stopped ($1k$ iterations), the obtained generator can neither produce realistic samples nor provide complete coverage of the real data. The previous accept-reject sampling methods applied to the generator at this stage can successfully reject the majority of the bad samples, but fail to recover the real distribution. In contrast, our collaborative sampling scheme succeeds in obtaining samples of both high quality and high diversity.

We next evaluate our method quantitatively, following the previous protocol in (Azadi et al. 2019; Turner et al. 2019). Samples that are less than four standard deviations away from the nearest Gaussian component are considered as *good*. We compute the KL divergence between the categorical distributions of real samples and good generated samples to measure the diversity of the good generated samples. To evaluate the overall performance, we introduce an extra category for the bad samples and compute the JS divergence between the augmented categorical distributions of real samples and all generated samples. As shown in Figure 4, our method exhibits superior performance in comparison to the existing sampling methods.

Figure 4d shows the scores of the diagnostic metrics as well as the JS divergence resulting from the MH algorithm using discriminators fine-tuned for different iterations. In addition to the Z-statistic promoted in (Turner et al. 2019), we also take the expected calibration error (ECE) (Naeini, Cooper, and Hauskrecht 2015), another popular metric for neural network calibration (Guo et al. 2017) into comparison. It is visually apparent that the evolvement of the Z-statistic is dramatically different from the other metrics. More detailed correlation coefficients between the diagnostic metrics and then performance gains from the rejection step are summarized in Table 1. Compared with the Z-statistic and ECE, the Brier Score exhibits a significantly stronger correlation with the performance gains, which validates the effectiveness of the Brier Score for discriminator diagnosis.

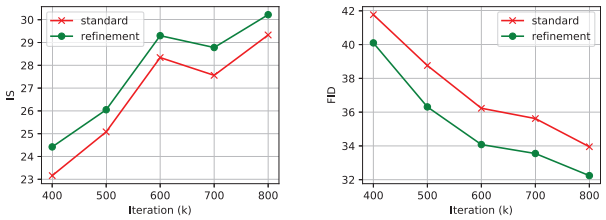
Image Generation

We next demonstrate the efficacy of our method in image generation tasks. In our experiments, we use the standard DCGAN (Radford, Metz, and Chintala 2015) for modelling



(a) CIFAR10 (b) CelebA

Figure 6: Quantitative comparison between our collaborative sampling scheme and baseline sampling methods for DCGAN on CIFAR10 and CelebA. Lower is better for FID.



(a) IS (b) FID

Figure 7: Quantitative comparison between our collaborative sampling scheme and the standard sampling for SAGAN on ImageNet (Batch size 16). Higher is better for IS and lower is better for FID.

the CIFAR10 (Krizhevsky 2009) and the CelebA (Liu et al. 2015) datasets, and the SAGAN (Zhang et al. 2019) for modelling ImageNet (Deng et al. 2009) at 128×128 resolution. For sample refinement, we conduct a maximum of 50 refinement steps with a step size of 0.1 in a middle layer of the generator for the DCGAN and 16 updates with a step size of 0.5 for the SAGAN. Performance is quantitatively evaluated using the Inception Score (IS) (Salimans et al. 2016) and the Fréchet Inception Distance (FID) (Heusel et al. 2017) on 50k images.

As shown in Figure 6 and Figure 7, our collaborative sampling scheme provides consistent performance boost at each training stage across different datasets and GAN variants, suggesting the strong ability of our method to improve the model distribution of complex data. In addition to the quantitative improvements, Figure 5 qualitatively compares the images proposed by the generator and those produced by our method on the CelebA dataset. The perceptual differences between the generated and refined images demonstrate the effectiveness of our method in identifying artifacts and improving image quality.

Image Manipulation

We next evaluate our collaborative sampling scheme with CycleGAN (Zhu et al. 2017), a popular method for image-to-image translation. To improve the image quality, we perform a maximum of 100 refinement steps. As shown in Figure 8, the modifications made by our method at 256×256 resolution concentrates on the pattern of the target class

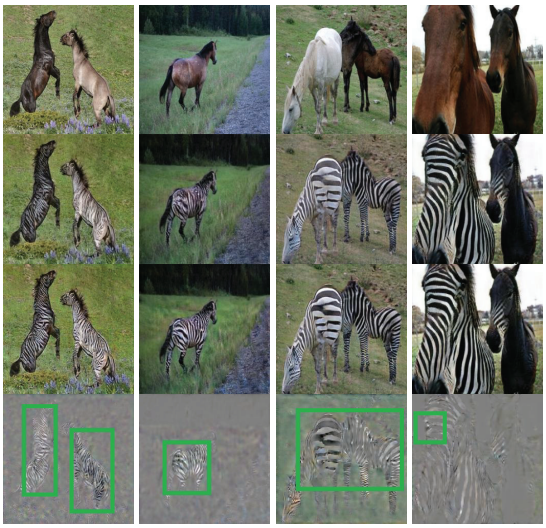


Figure 8: Results of our collaborative sampling scheme in CycleGAN for unpaired image-to-image translation at 256×256 resolution. The real horse images (top) are translated into synthetic zebra images (second row), which are further refined by our method (third row). The differences between the translated and refined images are concentrated on zebra patterns (bottom).

without affecting background semantics. This result validates the unique advantage of our method over the rejection algorithms for enhancing the output quality in the image manipulation tasks.

Key Attributes

We finally investigate the impact of two key attributes of our method through experiments on the MNIST (LeCun et al. 1998). Here, we use the original NS-GAN (Goodfellow et al. 2014) as a baseline and apply our collaborative sampling scheme for 20 refinement steps with a step size of 0.1.

Effect of Discriminator Shaping We highlight the importance of the proposed discriminator shaping by qualitatively comparing the MNIST images produced by three different sampling schemes: (a) standard GAN sampling (b) collaboratively sampling without discriminator shaping and (c) collaborative sampling with discriminator shaping. As shown in Figure 9, when the generated images contain small artifacts, the refinement process guided by the standard discriminator fails to remove the artifacts and instead adds more noises. In contrast, the shaped discriminator leads to visually more realistic digits. Figure 10 shows the quantitative effect of discriminator shaping on classifier score (CS) and Fréchet distance (FD), reaffirming the necessity of discriminator shaping.

Effect of Refinement Layer To examine the impact of the choice of the refinement layer, we visualize the difference between the refined samples and the originally proposed samples as a function of the layer index in Figure 11. The sample refinement performed at the output layer results in

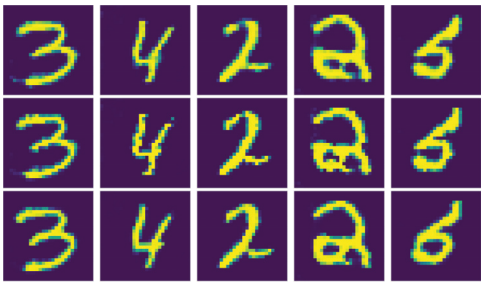
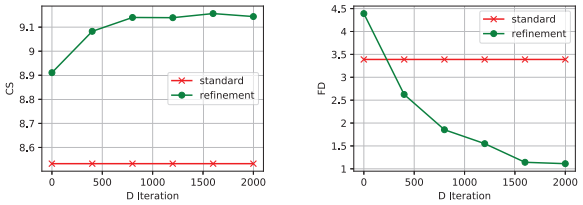


Figure 9: Qualitative effect of discriminator shaping. Refining images proposed by the generator *(first row)* using the standard discriminator without additional shaping leads to worse images *(second row)* in comparison to the images obtained after discriminator shaping *(third row)*.



(a) CS (b) FD

Figure 10: Quantitative effect of discriminator shaping. We apply our sample refinement method using discriminators shaped for different iterations. Higher is better for CS and lower is better for FD.

local modifications, whereas the refinement at the low-level activation map alters the global semantics. The choice of the middle layer leads to a balanced performance, fixing the local artifacts in "0" and "3" while making global changes in the other images that are far from being realistic.

Related Work

Designing collaborative mechanisms in addition to adversarial training has garnered growing interest in the past couple of years. (LeCun 2016) promoted to replace the discriminator by a collaborator to provide encouraging feedback. Recent works (Xie et al. 2018; Chen et al. 2019; Seddik, Tamaazousti, and Lin 2019) proposed concrete methods for training generative models collaboratively. In contrast, our work is focused on the design of collaboration mechanism during the sampling process.

To obtain desired samples, one line of work (Zhu et al. 2016; Nguyen et al. 2017; Yeh et al. 2016; Samangouei, Kabkab, and Chellappa 2018) employed gradient-based optimization in the latent space of GANs. The goal of these methods is essentially to seek samples from the learned data manifold closest to the target, whereas we aim to improve the learned data prior. By performing sample refinement at a selected layer of the generator, our method enables the refined samples to go beyond the data manifold modelled by the generator.

Another line of recent work proposed to modify the activation in a middle layer of the generator in order to manipulate samples with human intervention (Bau et al. 2019) or an additional neural network (Shama et al. 2019). On the contrary, our work provides a generic sample refinement method guided by the gradient from the discriminator, allowing one to exploit the full knowledge of the learned networks with theoretical inspirations.

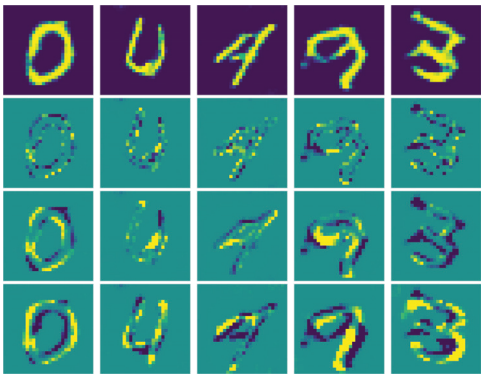


Figure 11: Qualitative effect of the choice of refinement layer. The last three rows show the differences between the generated sample (first row) and the refined samples when sample refinement (i) performed at the output layer (second row), (ii) the middle layer (third row), and (iii) the input of the generator (last row). We can observe sample modifications from micro to macro scales.

Our proposed discriminator shaping method can be viewed as a form of adversarial training, which is typically used to improve the robustness of a classifier (Madry et al. 2017). Recently, (Zhou and Krähenbühl 2019) proposed to train the discriminator adversarially in a restricted region for stabilizing the GAN training. Another concurrent work (Santurkar et al. 2019) demonstrated the generative power of a single adversarially robust classifier through extensive experiments. Our method adopts the adversarial training approach with the goal of guiding the collaborative sampling process effectively at test time.

Conclusions

We present a novel collaborative sampling scheme for using GANs at test time. Rather than disregarding the discriminator, we propose to continue using the gradients provided by a shaped discriminator to refine the generated samples. This is advantageous when the model distribution does not match the real data distribution. It is also highly valuable for applications where sample quality matters or the rejection sampling approach is not admissible. Orthogonal to existing techniques in GAN training, our method offers an additional degree of freedom to improve the generated samples empowered by the discriminator.

Acknowledgements

We thank Sven Kreiss and Tao Lin for helpful discussions. We also thank Taylor Mordan, Dhruti Shah for valuable feedback on drafts of this paper.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein GAN. *arXiv:1701.07875 [cs, stat]*. arXiv: 1701.07875.

Arora, S.; Ge, R.; Liang, Y.; Ma, T.; and Zhang, Y. 2017. Generalization and equilibrium in generative adversarial nets (gans). In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 224–232. JMLR. org.

Azadi, S.; Olsson, C.; Darrell, T.; Goodfellow, I.; and Odena, A. 2019. Discriminator rejection sampling. In *International Conference on Learning Representations*.

Bau, D.; Zhu, J.-Y.; Strobelt, H.; Bolei, Z.; Tenenbaum, J. B.; Freeman, W. T.; and Torralba, A. 2019. Gan dissection: Visualizing and understanding generative adversarial networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Brier, G. W. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review* 78(1):1–3.

Brock, A.; Donahue, J.; and Simonyan, K. 2019. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*.

Chen, T.; Zhai, X.; Ritter, M.; Lucic, M.; and Houlsby, N. 2019. Self-supervised gans via auxiliary rotation loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Dawid, A. P. 1997. Prequential analysis. *Encyclopedia of Statistical Sciences* 1:464–470.

Deng, J.; Dong, W.; Socher, R.; Li, L.; and and. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Fermi, E., and Richtmyer, R. 1948. Note on census-taking in monte-carlo calculations. Technical report, Los Alamos Scientific Lab., Los Alamos, NM.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A. C.; and Bengio, Y. 2014. Generative adversarial networks. *ArXiv* abs/1406.2661.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1321–1330. JMLR. org.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 6626–6637.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2017. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv:1710.10196 [cs, stat]*. arXiv: 1710.10196.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.; et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

LeCun, Y. 2016. Generative Collaborative Networks. *Twitter*.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, 3730–3738.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv:1706.06083 [cs, stat]*. arXiv: 1706.06083.

Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral Normalization for Generative Adversarial Networks. *arXiv:1802.05957 [cs, stat]*. arXiv: 1802.05957.

Murphy, A. H. 1973. A new vector partition of the probability score. *Journal of applied Meteorology* 12(4):595–600.

Naeini, M. P.; Cooper, G.; and Hauskrecht, M. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Neal, R. M. 1996. *Bayesian Learning for Neural Networks*. Berlin, Heidelberg: Springer-Verlag.

Nguyen, A.; Clune, J.; Bengio, Y.; Dosovitskiy, A.; and Yosinski, J. 2017. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4467–4477.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv:1511.06434 [cs]*. arXiv: 1511.06434.

Roberts, G. O., and Tweedie, R. L. 1996. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli* 2(4):341–363.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved Techniques for Training GANs. *arXiv:1606.03498 [cs]*. arXiv: 1606.03498.

Samangouei, P.; Kabkab, M.; and Chellappa, R. 2018. Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models. *arXiv:1805.06605 [cs, stat]*.

Santurkar, S.; Tsipras, D.; Tran, B.; Ilyas, A.; Engstrom, L.; and Madry, A. 2019. Computer vision with a single (robust) classifier. *ArXiv* abs/1906.09453.

Seddik, M. E. A.; Tamaazousti, M.; and Lin, J. 2019. Generative Collaborative Networks for Single Image Super-Resolution. *arXiv:1902.10467 [cs]*. arXiv: 1902.10467.

Shama, F.; Mechrez, R.; Shoshan, A.; and Zelnik-Manor, L. 2019. Adversarial feedback loop. In *Proceedings of the IEEE International Conference on Computer Vision*, 3205–3214.

Tierney, L. 1994. Markov chains for exploring posterior distributions. *the Annals of Statistics* 1701–1728.

Turner, R.; Hung, J.; Frank, E.; Saatchi, Y.; and Yosinski, J. 2019. Metropolis-hastings generative adversarial networks. In *International Conference on Machine Learning*, 6345–6353.

Xie, J.; Lu, Y.; Gao, R.; Zhu, S.; and Wu, Y. 2018. Cooperative training of descriptor and generator networks. *IEEE transactions on pattern analysis and machine intelligence*.

Yeh, R. A.; Chen, C.; Lim, T.-Y.; Schwing, A. G.; Hasegawa-Johnson, M.; and Do, M. N. 2016. Semantic image inpainting with deep generative models. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 6882–6890.

Zhang, H.; Goodfellow, I.; Metaxas, D.; and Odena, A. 2019. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, 7354–7363.

Zhou, B., and Krähenbühl, P. 2019. Don't let your discriminator be fooled. In *International Conference on Learning Representations*.

Zhu, J.-Y.; Krähenbühl, P.; Shechtman, E.; and Efros, A. A. 2016. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, 597–613. Springer.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232.