Collaborative Sampling in Generative Adversarial Networks

Yuejiang Liu*, Parth Kothari*, Alexandre Alahi
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
{firstname.lastname}@epfl.ch

Abstract

A standard practice in Generative Adversarial Networks (GANs) is to completely discard the discriminator when generating samples. 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 generated samples through gradient-based optimization 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 and further improve the sample refinement process. Through experiments on synthetic and image datasets, we demonstrate that our proposed method is able to improve generated samples both quantitatively and qualitatively, offering a new degree of freedom in GAN sampling. We finally show its potential for tackling mode collapse as well as adversarial examples.

1. Introduction

Generative Adversarial Networks (GANs) [1] are a class of deep generative models known for producing state-of-the-art realistic samples. Despite successful applications in a wide variety of tasks [1, 2, 3, 4, 5, 6, 7, 8], training GANs is notoriously unstable, impacting the model distribution. Numerous works have attempted to improve GAN training by introducing different loss functions [9, 10], regularization schemes [11, 12, 13], training procedures [14, 15] as well as novel architectures [2, 16]. Yet, stablizing GANs at scale remains an open problem. In this work, we go beyond the lines of work in modifying training process and explore methods for sampling process. Our goal is to improve the model distribution by fully exploiting the value contained in the trained networks.

*Equal contribution
Code is available at http://github.com/vita-epfl/collaborative-gan-sampling

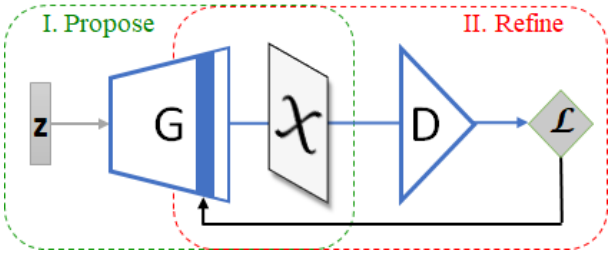


Figure 1: Once training completes, we use both the generator network and the discriminator network for collaborative sampling. Our scheme consists of one sample proposal step and multiple sample refinement steps. (I) The fixed generator proposes samples. (II) Subsequently, the discriminator provides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

A standard practice in GAN sampling is to completely discard the discriminator while using only the generator for sample generation. Recent works [17, 18] show that it is beneficial to post-process the generator distribution by rejecting bad samples based on the discriminator output. However, this accept-reject paradigm not only suffers from low sampling efficiency but also restricts the accepted samples to the data manifold learned by the generator. We address these limitations by refining, rather than simply rejecting, the generated samples.

Figure 1 illustrates our proposed collaborative sampling scheme between the generator and discriminator. Once training is complete, 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 the low-level feature maps to the final output space, until the samples are classified as "real" by the discriminator. Serving as a feedback loop, the sample refinement process shifts the model distribution closer to the real data distribution.

The performance of our collaborative sampling scheme is heavily dependent on the loss landscape provided by

the discriminator. To further improve the sample refinement process, we propose a practical discriminator shaping method that fine-tunes the discriminator using the refined samples. This method not only enhances the robustness of the discriminator for classification but also smoothens the learned loss landscape spanning from the generator distribution to the real data distribution, thereby strengthening the discriminator's ability to guide the sample refinement updates.

Through extensive experiments on both synthetic and real image datasets, we demonstrate that the proposed collaborative sampling scheme can consistently improve the generated samples both quantitatively and qualitatively, significantly outperforming other sampling methods. Moreover, we highlight that in challenging settings, where the standard GAN training is prone to mode collapse, our proposed framework has the potential to achieve high sample diversity without compromising the sample quality. Finally, we show that our method is not limited to improving generated samples by applying it to defend against adversarial examples. Our method can be applied on top of existing GAN training techniques, thereby offering a new degree of freedom to improve the generated samples through the sampling process.

2. Background

2.1. Generative Adversarial Networks

Generative Adversarial Networks (GANs) consist of two neural networks, namely the generator G and the discriminator D, trained together. The generator G takes as input a noise vector z sampled from a given noise distribution p_z and transforms it into a real looking sample $G(z)$. The discriminator D outputs a probability score indicating 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, which can be formulated as follows:

$$\min_G \max_D \mathbb{E}_{x\sim p_r}[\log(D(x))] + \mathbb{E}_{x'\sim p_g}[1 - \log(D(x'))]. \tag{1}$$

Goodfellow *et al.* [1] show that, under certain conditions, optimizing the above loss function leads to a generator exactly recovering the real data distribution.

2.2. GAN Training

Training GANs in practice is a notoriously challenging problem due to the complex dynamics of the minimax game. Without direct access to the real data, the generator learns only from the gradients provided by the discriminator. However, these gradients are prone to vanish [19] or explode [11], resulting in training instability. Goodfellow *et al.* [1] advocated the use of a non-saturating loss function (NS-GAN) to mitigate the issue of vanishing gradients:

$$\begin{aligned} \mathcal{L}_D &= -\mathbb{E}_{x\sim p_r}[\log D(x)] - \mathbb{E}_{z\sim p_z}[1 - \log D(G(z))] \\ \mathcal{L}_G &= -\mathbb{E}_{z\sim p_z}[\log D(G(z))]. \end{aligned} \tag{2}$$

To further improve GAN training, recent works have been extensively focused on various training procedures [15, 20], loss functions [10], network architectures [2], regularization [11, 12] and normalization methods [13]. Despite improved stability, these techniques can overly constrain the discriminator resulting in a trade-off between training stability and algorithmic performance. Recent work [3] achieved state-of-the-art results by relaxing these stabilizing conditions and allowing training collapse. Moreover, [21, 22] show that with proper hyperparameter tuning, the original non-saturating GAN (Eq. 2) can achieve comparable performance to the other latest variants. In this work, we sidestep these training issues. Instead, we accept the trained generator distribution at its face value and try to improve the generated samples during the sampling process.

2.3. GAN Sampling

The standard GAN sampling process draws samples from the generator distribution without the involvement of the discriminator. Recently, [17] proposed a rejection sampling scheme using the discriminator to filter out samples that are unlikely to be real. [18] replaced the rejection sampling by the Markov chain Monte Carlo (MCMC) method for better scalability in high-dimensional space. However, both these methods rely on an accept-reject principle and inevitably sacrifice sample efficiency and flexibility. For practical usage, these methods have to modulate the rejection threshold [17] or calibrate the discriminator [18] to counter the low acceptance rate. Our work explores a more involved collaboration between the generator and the discriminator which exploits higher-order information contained in the discriminator to improve the generated samples.

3. Related Work

Designing collaborative mechanisms in addition to the adversarial training has garnered growing interest for deep generative models. LeCun [23] promoted to replace the discriminator by a collaborator to provide encouraging feedback. A conceptually unadversarial framework is jokingly introduced in [24]. Several recent works present more concrete realizations of the collaborative paradigm. Chen *et al.* [25] proposed a method for the generator and discriminator to collaborate on representation learning while competing on the generative task. A similar work [26] trained the generator and the discriminator collaboratively to learn relevant features for super-resolution. In contrast to these methods,

our work is focused on collaboration mechanisms during sampling process.

GAN-based sample modifications have been recently explored in several tasks, such as image editing [27], image inpainting [28], super-resolution [29], adversarial defenses [30]. Most of these works update the prior of the generator in the latent space to seek an output closest to their target. Another recent work proposed to activate or deactivate a part of the middle layer of the generator with the involvement of human intervention [31]. Our work provides a more generic framework guided by the discriminator, which allows the sample refinement to be performed any layer of the generator and enables the refined samples to go beyond the data manifold learned by the generator.

Our proposed discriminator shaping method can be viewed as a form of adversarial training, which is widely used in classification tasks to improve robustness [32, 33, 34]. Concurrent to our work, Zhou *et al.* [35] proposed to train the discriminator adversarially in a restricted region around the generated samples in order to stabilize the GAN training. In our work, we shape the discriminator loss landscape in a wider range spanning from the generator distribution to the real distribution in order to more effectively guide the sample refinement process.

4. Method

In this section, we present a GAN sampling method that uses both the generator and the discriminator to produce samples collaboratively. In addition, we introduce a discriminator shaping method that smoothens the discriminator loss landscape and further improves the collaborative sampling process.

4.1. 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:

$$\begin{aligned} G(z) &= G_L \circ G_{L-1} \circ \cdots \circ G_1(z), \\ G_l(x_l) &= \sigma(\theta_l \cdot x_l) + b_l, \quad l = 1, 2, \ldots, L, \end{aligned} \tag{3}$$

where G_l is the lth layer of the generator, x_l is the activation input, σ is a nonlinear activation function, θ_l and b_l are the model parameters. The input to the first layer $x_1 = z$ and the output of the last layer $G_L(x_L) = x$. 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 real samples, *i.e.*, $D(x) = 1/2$ and $\frac{\partial \mathcal{L}_G}{\partial x} = 0$ (see Eq. 2). However, such a saddle point of the minimax problem is hardly recovered in practice [36], indicating room for improving p_g.

Algorithm 1 Collaborative Sampling

Input: a frozen generator G, a frozen discriminator D, the index of layer for sample refinement l, the maximum steps of sample refinement K, the average discriminator output for real samples $\tilde{D}$
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 $D(x^k) < \tilde{D}$ **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. 4)
    $x^{k+1} \leftarrow \text{UpdateSample}(G, x_l^{k+1})$, (Eq. 5)
  else
    break
  end if
end for

Our goal is to shift p_g towards p_r through post-processing without changing network parameters. Inspired by recent works [17, 18] which reject undesired samples according to $D(x)$, we leverage the gradient information provided by the discriminator to continuously refine the generated samples through the following recursive update:

$$x_l^{k+1} = x_l^k - \lambda \nabla_l \mathcal{L}_G(x_l^k), \tag{4}$$

$$x^{k+1} = G_L \circ G_{L-1} \circ \ldots G_l(x_l^{k+1}), \tag{5}$$

where k is the iteration number, λ is the stepsize, l is the index of the generator layer for sample refinement. The recursion consists of two parts: in the backward pass, the discriminator provides the generator with feedback gradients to adjust the activation map of the selected layer l (Eq. 4); in the forward pass, the generator reuses part of its parameters to propose an improved sample (Eq. 5). This method forms a closed-loop sampling process, allowing both the generator and the discriminator to contribute to sample generation collaboratively. A pseudo code is summarized in Algorithm 1.

4.2. Discriminator Shaping

The effectiveness of the proposed collaborative sampling scheme highly depends on the loss landscape provided by the discriminator. In the case of a smooth and monotonic loss landscape from the generator distribution p_g to the real distribution p_r, every generated sample can be refined to the real distribution according to the gradient feedback. However, this is not always the case. In the standard GAN training, the discriminator is trained solely to distinguish between the real and fake samples, and is thus prone to overfit to the generator distribution. As a consequence, the discriminator may misclassify a poorly refined sample and fail to suggest further improvements.

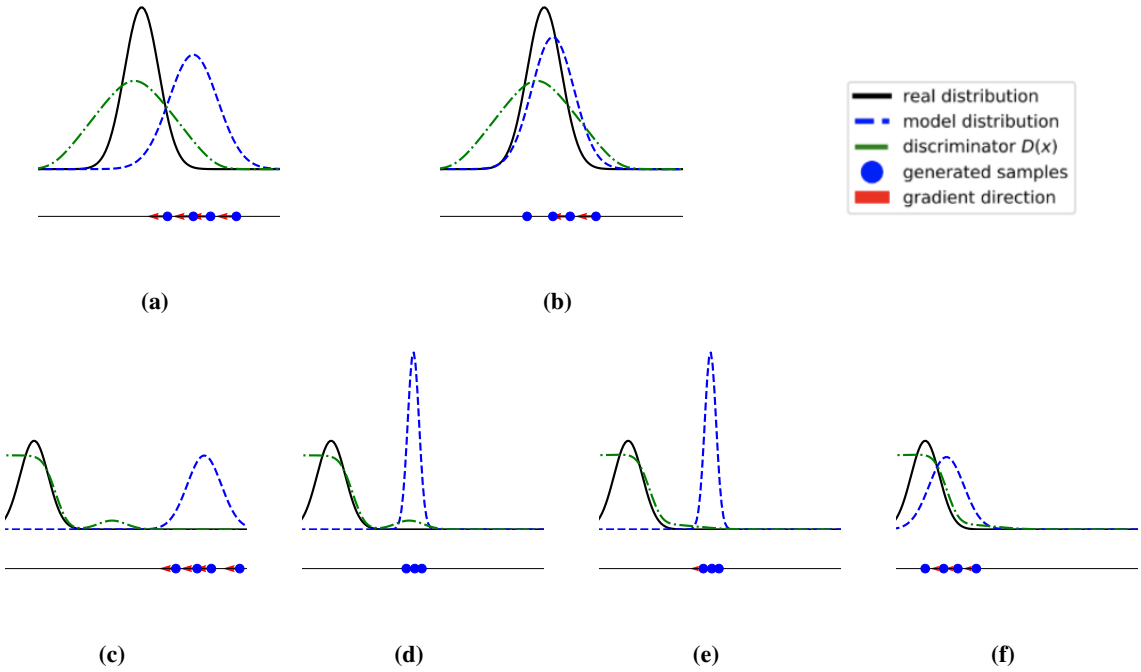


Figure 2: (a)-(b) illustrates our collaborative sampling scheme in the ideal scenario. The samples from the generator distribution are refined using the gradient information provided by the discriminator, resulting in a model distribution closer to the real distribution. (c)-(f) Illustrates our collaborative sampling scheme in the non-ideal scenario. (c)-(d) Illustrates the refined samples getting stuck in a bad local optima due to the complex discriminator loss landscape. (e) Discriminator shaping is performed using refined samples to smoothen the loss landscape. (f) The new loss landscape helps to better approximate the real distribution from the generator distribution.

To resolve this issue, we devise a practical discriminator shaping method in order to obtain a discriminator that is not only accurate in classifying the generated samples but also capable of effectively guiding the sample refinement process. Given a trained generator and discriminator, our method fine-tunes the discriminator to optimize the following modified objective:

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

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

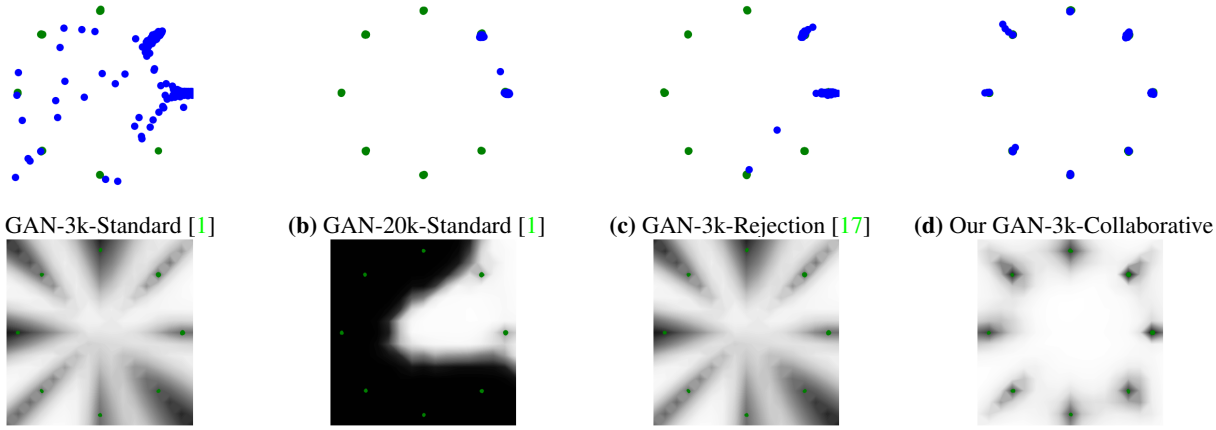
As outlined in Algorithm 2, we conduct the discriminator shaping and collaborative sampling alternatively, aiming to remove poor local optima and smoothen the loss landscape learned by the discriminator. It is essentially a self-supervised post-training procedure, gradually pushing the discriminator to generalize and collaborate with the generator for sample refinement.

Algorithm 2 Discriminator Shaping

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

4.3. Discussion

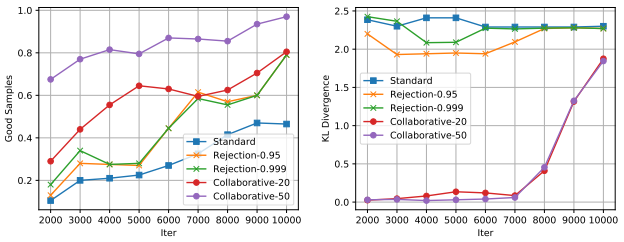
A key hyperparameter in our collaborative sampling scheme is the index of the generator layer l for sample refinement. On one extreme, we can adjust the proposed sample at the output of the generator, which is equivalent to modifying the sample directly. Manipulating a proposed sample in the data space does not rely on any part of the generator and thus, in principle, can result in an arbitrarily good replacement. However, shifting a high-dimensional sample such as a natural image from a low-density region to a high-density region often requires a large number of iterations, which is not desirable for real-time sample generation. On the other extreme, we can choose to adjust the latent code z. The dimension of the latent code is normally much smaller



(a) GAN-3k-Standard [1] **(b)** GAN-20k-Standard [1] **(c)** GAN-3k-Rejection [17] **(d)** Our GAN-3k-Collaborative

(e) GAN-3k-Standard [1] **(f)** GAN-20k-Standard [1] **(g)** GAN-3k-Rejection [17] **(h)** Our GAN-3k-Collaborative

Figure 3: Qualitative evaluation of collaborative sampling in GANs on a synthetic imbalanced mixture of 8 Gaussians dataset. *(a)* Terminating GAN training early results in bad accuracy while *(b)* training GANs till convergence results in mode collapse. Our collaborative sampling scheme *(d)* applied to an early terminated GAN is successful in recovering all modes without compromising sample quality, significantly outperforming (c) the rejection sampling method [17]. *(e)-(h)* Visualization of the discriminator loss landscape. The darker the region, the higher the discriminator score. The rejection sampling method (g) does not change the discriminator, whereas our method (h) leads to a better shaped discriminator loss landscape.



(a) Quality **(b)** Diversity

Figure 4: Quantitative comparison of different sampling methods on the imbalanced mixture of Gaussians dataset. (a) % of good samples (within a distance of 3 times standard deviation from the nearest mode): Higher is better. *(b)* KL Divergence between the real distribution and model distribution (each good sample is assigned to the nearest mode): Lower is better.

than the dimension of the sample output. Therefore, refining a sample in the latent space is computationally more efficient. The downside is that the resulting samples are restricted to the data distribution defined by the generator. We empirically find that refining a sample at a middle layer of the generator leads to a good balance between the computational efficiency and flexibility for collaborative sampling.

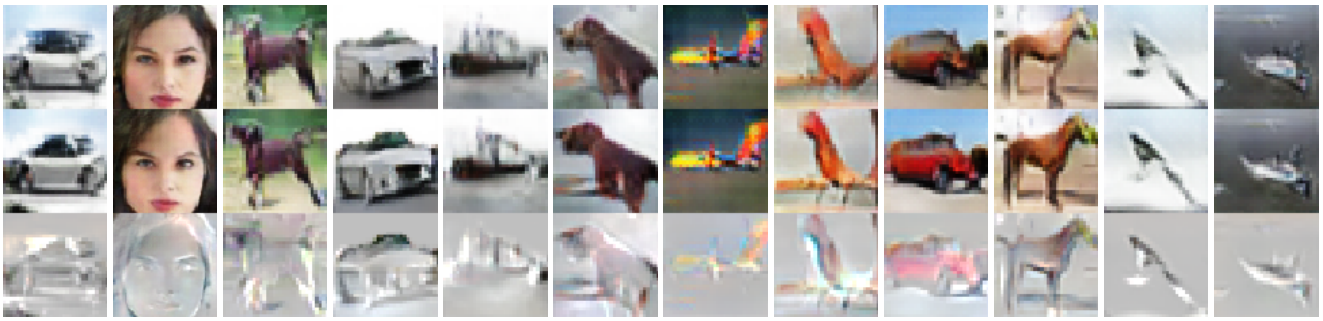
Figure 2 illustrates the proposed methods in a 1D scenario. As shown in Figure (2a), a trained generator provides a model distribution that is close to, but not exactly the same as, the real data distribution. When the discriminator loss function is sufficiently smooth and monotonic (Figure (2b), the collaborative sampling scheme can easily shift the generator distribution closer to the real one. However, in high dimensional space, the loss function provided by the default discriminator may present numerous local optima. As a result, the sample refinement may stop in regions that are still distant from the real samples. Figure (2e) and Figure (2f) show how recursively shaping the discriminator using the refined data distribution can help in better approximating the real data distribution.

Our sampling scheme provides improved sample quality but at the expense of extra iterations. These additional computational costs not only depend on the refinement layer and optimization algorithms but also reflect the quality gap between the proposed samples and refined ones. In the next section, we show experimentally that considerable improvements in sample quality can be obtained within 20 to 50 refinement steps.

5. Experiments

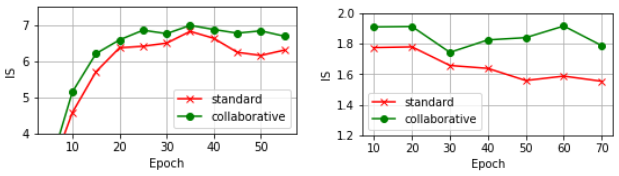
In this section, we present empirical results to validate the proposed collaborative sampling scheme. We first compare our method with other sampling methods on several datasets both quantitatively and qualitatively. Additionally, we highlight the potential of our method in addressing mode collapse, a common pitfall in standard GAN training. Further, we examine the effect of discriminator shaping as well as the choice of the refinement layer. Finally, we investigate the potential of our method to refine samples for adversarial defense.

(a) CelebA



(b) CIFAR10

Figure 5: Qualitative illustration of the performance of our collaboratively sampling in GANs on the CelebA (a) and CIFAR10 (b). Both datasets are modeled using the DCGAN [2] trained for 30 epochs. The discriminator is shaped for 1 epoch and activation map of second generator layer is refined for 20 steps. *(Top)* Samples produced by using only the generator [1]. *(Middle)* Samples produced by our collaboratively sampling method. *(Bottom)* The difference between the top and middle row images for visualizing the refinement.



(a) CIFAR10 **(b)** CelebA

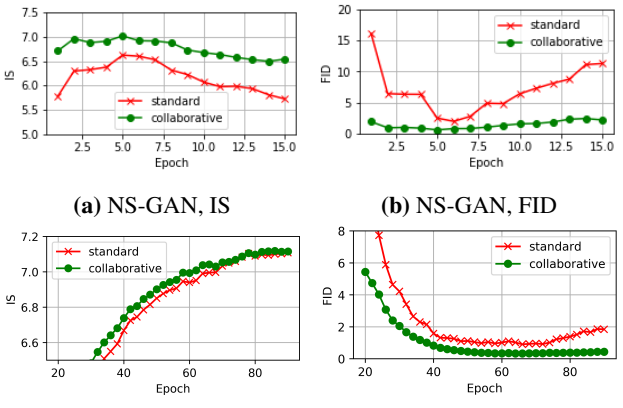
Figure 6: Quantitative comparison between our collaborative sampling scheme and the standard GAN sampling [1] on CIFAR10 and CelebA. Higher is better for IS

5.1. Synthetic Data

We first assess our collaborative sampling scheme on a 2D synthetic dataset. The dataset consists of an imbalanced mixture of Gaussians, in which 90% of the real samples are from two Gaussian components while the rest 10% are from the other six. We use a standard fully-connected MLP with 4 hidden layers to model the generator and the discriminator. We shape the discriminator for 5k additional iterations after terminating the standard GAN training.

Figure 3 shows the qualitative results of our proposed method as well as the other baselines. The standard GAN training gradually runs into mode collapse on the imbalanced dataset, resulting in high sample quality but poor diversity after 20k iterations. If early stop the GAN training, the generator at 3k iterations is not able to produce realistic samples. The rejection sampling method applied to the generator at early stage can boost the sample quality but fail to maintain high diversity. In contrast, by applying the collaborative sampling method to the same generator, we succeed in obtaining samples of both high quality and diversity. It is also visually clear that the loss landscape after discriminator shaping has a high similarity with that of the real Gaussian components.

Figure 4 provides a quantitative comparison of different sampling methods applied at various training stages. Our method with 20 steps of refinement achieves highly competitive results on sample quality while obtaining significantly better diversity than the standard sampling and the rejection sampling method with different threshold parameters



(a) NS-GAN, IS **(b)** NS-GAN, FID

(c) WGAN-GP, IS **(d)** WGAN-GP, FID

Figure 7: Quantitative comparison between our collaborative sampling scheme and the standard GAN sampling [1] on an imbalanced MNIST dataset with NS-GAN [1] and WGAN-GP [11]. Higher is better for IS and lower is better for FID.

[17]. By further increasing the number of refinement steps, our method outperforms the other baselines by a significant margin on both metrics.

5.2. Image Generation

We next demonstrate the effectiveness of our method on image generation tasks. In our experiments, we use the standard DCGAN [2] for modeling the CIFAR10 [37] and the CelebA [38] datasets and use the NS-GAN [1] or the WGAN-GP [11] for the MNIST [39] dataset. We train both the generator and discriminator using the Adam optimizer [40] with a learning rate of 0.0002. For sample refinement, we conduct maximum 50 updates with a step size of 0.1 in a middle layer of the generator. Performance is quantitatively evaluated with the Inception Score (IS) [19] and the Fréchet Inception Distance (FID) [41].

Figure 5 compares the images proposed by the generator and produced by our collaborative sampling method on the CelebA and CIFAR10 dataset. We highlight the difference between the default samples and refined samples to reflect the capability of our method in improving the quality of natural images.

In addition to the qualitative improvements, we plot the quantitative results for image generation in Figure 6 and Figure 7. Our sampling scheme provides consistent performance boost for each training stage across different datasets and GAN variants, empirically suggesting the strong ability of our method to improve the model distribution of complex data.

5.3. Key Attributes of Collaborative Sampling

We further investigate the effect of two key aspects of our proposed collaborative sampling scheme: the discriminator shaping process and the choice of the refinement layer.

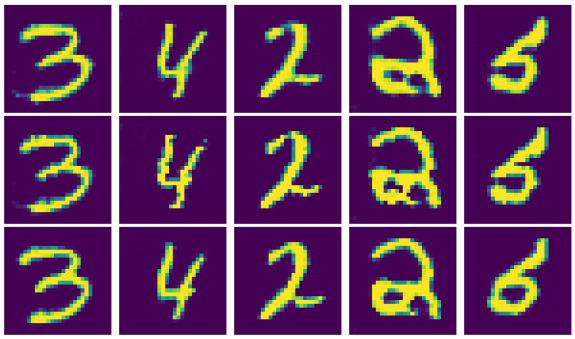


Figure 8: Effect of discriminator shaping: Refining images proposed by the generator *(first row)* using the standard discriminator without additional shaping leads to worse images *(second row)* in comparison to the images obtained after discriminator shaping *(third row)*.

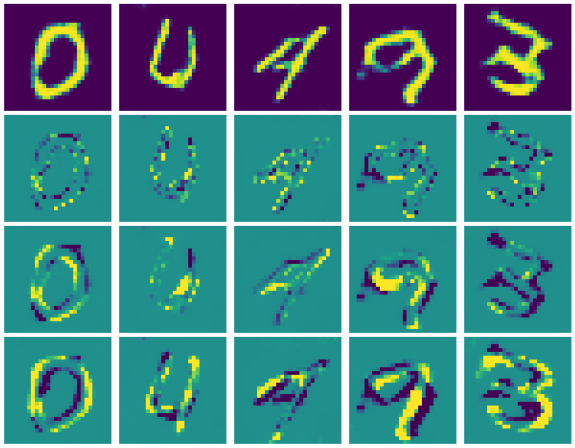


Figure 9: Effect of refinement layer. The generated samples *(first row)* are refined for 20 steps with a step size of 0.1. The refinement performed at the output *(second row)*, the middle layer *(third row)* and the input *(fourth row)* of the generator leads to sample modifications from micro to macro scale.

5.3.1 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 8, when the generated images contain small artifacts, the sample refinement process guided by the standard discriminator fails to remove these artifacts and instead add more noises. In contrast, the shaped discriminator leads to significantly better images by intensifying selected pixels to form more realistic images as well as removing the noise wherever necessary.

Norm	L1	L2	L^∞
z-space	10.11	99.51	1.77
hidden layer 1	3.28	25.06	1.00
hidden layer 2	2.06	13.80	0.77
hidden layer 3	1.76	11.81	0.64
x-space	0.63	9.92	0.14

Table 1: Average distances between the generated samples and the refined samples at different layers. Measured on 10000 instances for 10 refinement steps with the stepsize of 1.0.

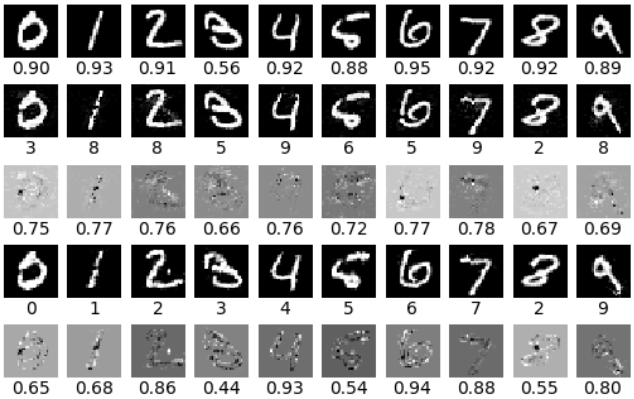


Figure 10: Defense against adversarial examples using discriminator guided sample refinement. A classifier is highly accurate on legitimate images *(first row)*, but completely fooled by adversarial examples [43] *(second row)*. Through 50 steps of sample refinement in the data space, our method provides cleaner images *(fourth row)*. The noises on adversarial images *(third row)* and refined images *(fifth row)* are presented for better visualization. Integer numbers indicates the predicted digits while floating point numbers indicate the probability associated with the predicted class.

5.3.2 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 9. We observe that the sample refinement performed at the output layer results in very local modifications. In contrast, the refinement at the low-level activation map alters the global semantics, even changing some of the digits. The refinement at the middle layer demonstrates a more balanced performance, fixing the local artifacts in "0" and "3" while making global changes to the other images that are far from real. Table 1 reports the average changes for the sample refinement at different layers, which confirms that adjusting the lower level activation can accelerate the overall changes (L1 and L2 norm) but tends to affect visual semantics (L^∞) [42].

Legitimate	Attack	Refined Adv.	Refined Leg.
0.9938	0.00	0.9531	0.9844

Table 2: Average classification accuracy for the legitimate and adversarial examples with or without defenses through sample refinement. Evaluated on 10000 instances.

5.4. Adversarial Defense

Motivated by the power of our method in refining generated samples, we finally investigate the potential of our method for improving other out-of-distribution samples, for instance, adversarial examples that an attacker intentionally designs to cause misprediction [44]. To this effect, we first train a classifier that achieves high accuracy on MNIST. We then construct untargeted adversarial examples by a strong optimization-based adversary [43], which can completely fool the classifier into making confident errors. To counter these malicious perturbations, we refine the input samples using the discriminator in Sec. 5.2, aiming to shift the adversarial examples towards the training distribution. As shown in Figure 10, most of the samples after refinement are correctly classified. One exception is the digit '8', which is still misclassified as '2'. However, it is interesting to note that the confidence assigned by the classifier for this image is comparably low, indicating a potential prediction error. The average accuracy before and after sample refinement is reported in Table 2. Refining the input with our method lifts the accuracy from 0% to 95% for adversarial examples, at the cost of a less than 1% accuracy drop for legitimate examples.

6. Conclusions

We present a collaborative sampling scheme in GANs. Rather than disregarding the discriminator during sampling, we propose to continue using the gradients provided by a shaped discriminator to refine the generated samples. This is advantageous in cases where the generator distribution does not exactly match the real data distribution. Orthogonal to various works in GAN training, our method offers another degree of freedom to improve the generated samples empowered by the discriminator.

The potential of our method is not limited to GAN sampling, as seen by its application to adversarial defenses. We hope to further analyze its potential for tackling mode collapse under various setups and explore its applications to the scenarios where rejection sampling is not admissible, such as image-to-image translations [5] and super-resolution [45], in the future.

Acknowledgments

We thank Sven Kreiss, Tao Lin, Su Li for valuable discussions. We also thank Dhruti Shah, Lorenzo Bertoni,

George Adaimi, Saeed Saadatnejad for helpful feedback on drafts of this paper.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv:1511.06434 [cs]*, Nov. 2015, arXiv: 1511.06434. [Online]. Available: http://arxiv.org/abs/1511.06434

[3] A. Brock, J. Donahue, and K. Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis," *arXiv:1809.11096 [cs, stat]*, Sep. 2018, arXiv: 1809.11096. [Online]. Available: http://arxiv.org/abs/1809.11096

[4] C. Ledig, L. Theis, F. Huszr, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 105–114.

[5] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," *arXiv:1703.10593 [cs]*, Mar. 2017, arXiv: 1703.10593. [Online]. Available: http://arxiv.org/abs/1703.10593

[6] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 5967–5976.

[7] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," *arXiv:1812.04948 [cs, stat]*, Dec. 2018, arXiv: 1812.04948. [Online]. Available: http://arxiv.org/abs/1812.04948

[8] Y. Chen, M. W. Hoffman, S. G. Colmenarejo, M. Denil, T. P. Lillicrap, M. Botvinick, and N. de Freitas, "Learning to Learn without Gradient Descent by Gradient Descent," *arXiv:1611.03824 [cs, stat]*, Nov. 2016, arXiv: 1611.03824. [Online]. Available: http://arxiv.org/abs/1611.03824

[9] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least Squares Generative Adversarial Networks," Nov. 2016. [Online]. Available: https://arxiv.org/abs/1611.04076

[10] J. Wu, Z. Huang, J. Thoma, D. Acharya, and L. Van Gool, "Wasserstein Divergence for GANs," *arXiv:1712.01026 [cs]*, Dec. 2017, arXiv: 1712.01026. [Online]. Available: http://arxiv.org/abs/1712.01026

[11] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved Training of Wasserstein GANs," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5767–5777. [Online]. Available: http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf

[12] N. Kodali, J. Abernethy, J. Hays, and Z. Kira, "On Convergence and Stability of GANs," *arXiv:1705.07215 [cs]*, May 2017, arXiv: 1705.07215. [Online]. Available: http://arxiv.org/abs/1705.07215

[13] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral Normalization for Generative Adversarial Networks," *arXiv:1802.05957 [cs, stat]*, Feb. 2018, arXiv: 1802.05957. [Online]. Available: http://arxiv.org/abs/1802.05957

[14] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, "Stacked Generative Adversarial Networks," *arXiv:1612.04357 [cs, stat]*, Dec. 2016, arXiv: 1612.04357. [Online]. Available: http://arxiv.org/abs/1612.04357

[15] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," *arXiv:1710.10196 [cs, stat]*, Oct. 2017, arXiv: 1710.10196. [Online]. Available: http://arxiv.org/abs/1710.10196

[16] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-Attention Generative Adversarial Networks," *arXiv:1805.08318 [cs, stat]*, May 2018, arXiv: 1805.08318. [Online]. Available: http://arxiv.org/abs/1805.08318

[17] S. Azadi, C. Olsson, T. Darrell, I. Goodfellow, and A. Odena, "Discriminator Rejection Sampling," *arXiv:1810.06758 [cs, stat]*, Oct. 2018, arXiv: 1810.06758. [Online]. Available: http://arxiv.org/abs/1810.06758

[18] R. Turner, J. Hung, Y. Saatci, and J. Yosinski, "Metropolis-Hastings Generative Adversarial Networks," *arXiv:1811.11357 [cs, stat]*, Nov. 2018, arXiv: 1811.11357. [Online]. Available: http://arxiv.org/abs/1811.11357

[19] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved Techniques for Training GANs," *arXiv:1606.03498 [cs]*, Jun. 2016, arXiv: 1606.03498. [Online]. Available: http://arxiv.org/abs/1606.03498

[20] T. Chavdarova and F. Fleuret, "SGAN: An Alternative Training of Generative Adversarial Networks," *arXiv:1712.02330 [cs, stat]*, Dec. 2017, arXiv: 1712.02330. [Online]. Available: http://arxiv.org/abs/1712.02330

[21] W. Fedus, M. Rosca, B. Lakshminarayanan, A. M. Dai, S. Mohamed, and I. Goodfellow, "Many Paths

to Equilibrium: GANs Do Not Need to Decrease a Divergence At Every Step," *arXiv:1710.08446 [cs, stat]*, Oct. 2017, arXiv: 1710.08446. [Online]. Available: http://arxiv.org/abs/1710.08446

[22] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are GANs Created Equal? A Large-Scale Study," *arXiv:1711.10337 [cs, stat]*, Nov. 2017, arXiv: 1711.10337. [Online]. Available: http://arxiv.org/abs/1711.10337

[23] Y. LeCun, "Generative Collaborative Networks," *Twitter*, 2016. [Online]. Available: https://twitter.com/boredyannlecun/status/791115429766766592?lang=en

[24] S. Albanie, S. Ehrhardt, and J. F. Henriques, "Stopping GAN Violence: Generative Unadversarial Networks," *arXiv:1703.02528 [cs, stat]*, Mar. 2017, arXiv: 1703.02528. [Online]. Available: http://arxiv.org/abs/1703.02528

[25] T. Chen, X. Zhai, M. Ritter, M. Lucic, and N. Houlsby, "Self-Supervised Generative Adversarial Networks," *arXiv:1811.11212 [cs, stat]*, Nov. 2018, arXiv: 1811.11212. [Online]. Available: http://arxiv.org/abs/1811.11212

[26] M. E. A. Seddik, M. Tamaazousti, and J. Lin, "Generative Collaborative Networks for Single Image Super-Resolution," *arXiv:1902.10467 [cs]*, Feb. 2019, arXiv: 1902.10467. [Online]. Available: http://arxiv.org/abs/1902.10467

[27] J.-Y. Zhu, P. Krhenbhl, E. Shechtman, and A. A. Efros, "Generative Visual Manipulation on the Natural Image Manifold," *arXiv:1609.03552 [cs]*, Sep. 2016, arXiv: 1609.03552. [Online]. Available: http://arxiv.org/abs/1609.03552

[28] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic Image Inpainting With Deep Generative Models," 2017, pp. 5485–5493. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2017/html/Yeh_Semantic_Image_Inpainting_CVPR_2017_paper.html

[29] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep Image Prior," *arXiv:1711.10925 [cs, stat]*, Nov. 2017, arXiv: 1711.10925. [Online]. Available: http://arxiv.org/abs/1711.10925

[30] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models," *arXiv:1805.06605 [cs, stat]*, May 2018, arXiv: 1805.06605. [Online]. Available: http://arxiv.org/abs/1805.06605

[31] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba, "GAN Dissection: Visualizing and Understanding Generative Adversarial Networks," *arXiv:1811.10597 [cs]*, Nov. 2018, arXiv: 1811.10597. [Online]. Available: http://arxiv.org/abs/1811.10597

[32] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv:1312.6199 [cs]*, Dec. 2013, arXiv: 1312.6199. [Online]. Available: http://arxiv.org/abs/1312.6199

[33] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," *arXiv:1706.06083 [cs, stat]*, Jun. 2017, arXiv: 1706.06083. [Online]. Available: http://arxiv.org/abs/1706.06083

[34] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan, "Theoretically Principled Trade-off between Robustness and Accuracy," *arXiv:1901.08573 [cs, stat]*, Jan. 2019, arXiv: 1901.08573. [Online]. Available: http://arxiv.org/abs/1901.08573

[35] B. Zhou and P. Krhenbhl, "Don't let your Discriminator be fooled," Sep. 2018. [Online]. Available: https://openreview.net/forum?id=HJE6X305Fm

[36] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and Equilibrium in Generative Adversarial Nets (GANs)," *arXiv:1703.00573 [cs, stat]*, Mar. 2017, arXiv: 1703.00573. [Online]. Available: http://arxiv.org/abs/1703.00573

[37] A. Krizhevsky, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

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

[39] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[40] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Dec. 2014, arXiv: 1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980

[41] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," *arXiv:1706.08500 [cs, stat]*, Jun. 2017, arXiv: 1706.08500. [Online]. Available: http://arxiv.org/abs/1706.08500

[42] I. Goodfellow, "Defense Against the Dark Arts: An overview of adversarial example security research and future research directions," *arXiv:1806.04169 [cs, stat]*, Jun. 2018, arXiv: 1806.04169. [Online]. Available: http://arxiv.org/abs/1806.04169

[43] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," *arXiv:1608.04644 [cs]*, Aug. 2016, arXiv: 1608.04644. [Online]. Available: http://arxiv.org/abs/1608.04644

[44] T. B. Brown, N. Carlini, C. Zhang, C. Olsson, P. Christiano, and I. Goodfellow, "Unrestricted Adversarial Examples," *arXiv:1809.08352 [cs, stat]*, Sep. 2018, arXiv: 1809.08352. [Online]. Available: http://arxiv.org/abs/1809.08352

[45] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.