# IMPROVING GENERATIVE ADVERSARIAL NETWORKS VIA ADVERSARIAL LEARNING IN LATENT SPACE

**Anonymous authors**
Paper under double-blind review

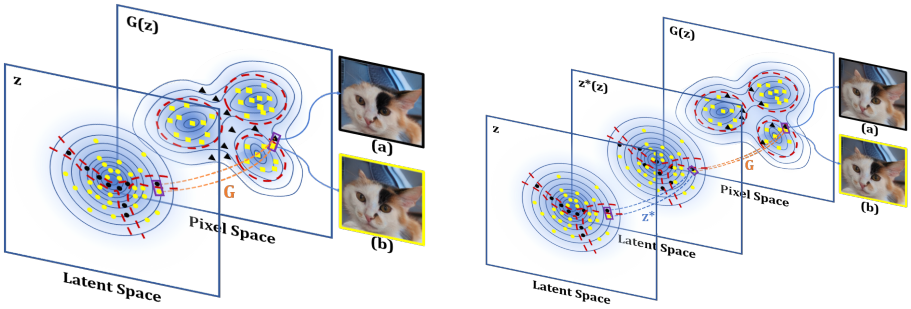
## ABSTRACT

Generative Adversarial Networks (GANs) have been widely studied as generative models, which map a latent distribution to the target distribution. Although many efforts have been made in terms of backbone architecture design, loss function, and training techniques, few results have been obtained on how the sampling in latent space can affect the final performance, and existing works on latent space mainly focus on controllability. We make an observation that, as the neural generator is a continuous function, two close samples in latent space would be mapped into two nearby images, while their quality can differ much as the quality is not a continuous function in pixel space. From the above continuous mapping function perspective, it is also possible that two distant latent samples are mapped into two close images. If the latent samples are mapped in aggregation into limited modes or even a single mode, mode collapse occurs. Accordingly, we propose adding an implicit latent transform before the mapping function to improve latent $z$ from its initial distribution, e.g., Gaussian. This is achieved by using the iterative fast gradient sign method (I-FGSM). We further propose a new training strategy named AdvLatGAN (Adversarial Latent GAN) for training the generator to obtain a better generation mapping by introducing targeted latent transforms into the bi-level optimization of GAN. Experimental results on visual data show that our method can effectively achieve improvement in both generation quality and diversity.

## 1 INTRODUCTION

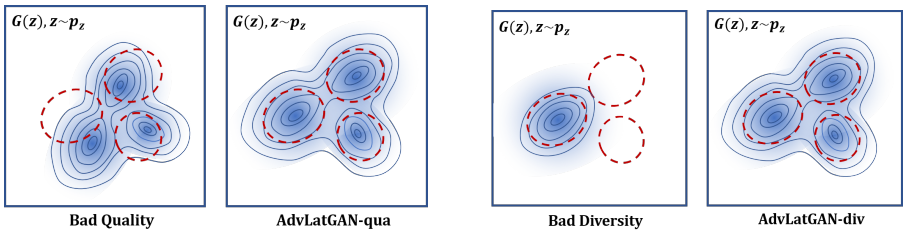
Generative Adversarial Networks (GANs) (Goodfellow et al., 2014a) have shown effectiveness for generating high-fidelity data, especially for images under different settings (Ledig et al., 2017; Wang et al., 2018; Zhu et al., 2017; Zhan et al., 2019; Chakraborty et al., 2018). Based on the zero-sum game, the model is trained by the adversarial process between the generator and the discriminator. Many efforts have been made to achieve more realistic generation from different perspectives. WGAN (Arjovsky et al., 2017), SNGAN (Miyato et al., 2018), LSGAN (Mao et al., 2017) aim to design better objective functions. ImprovedGAN (Salimans et al., 2016), AC-GAN (Odena et al., 2017) propose practical training techniques. While powerful network structures are studied in BigGAN (Brock et al., 2018), StyleGAN (Karras et al., 2019), SAGAN (Zhang et al., 2019).

Despite the above progress, an intriguing and relatively less studied question is how the latent space affects the generation in terms of both quality and diversity, which can be orthogonal to the above frequently studied factors like model architectures and training techniques. Specifically, we argue that the latent samples from a standard continuous distribution (e.g. Gaussian) can often be mapped to varying-quality samples for the generation. The reason is that the generator is a continuous function (by neural nets) while the generation quality (e.g. images in pixel space) is not. Moreover, it has also been shown in GAN literature (Khayatkhoei et al., 2018) that many image objects lie upon multiple disjoint manifolds, and a continuous mapping from continuous latent distribution e.g. Gaussian must generate many invalid samples in the between of the manifolds in order to cover all the meaningful modes. See Fig. 1 (a) for schematic diagram. This fact also suggests a dilemma: for complex real-world data, there is a tradeoff between quality and diversity when the latent vectors are sampled from a simple uni-mode distribution e.g. Gaussian, and the generator is a continuous mapping based on neural networks. Moreover, partly due to the mapping function that allows for many-to-one mapping, mode collapse (Srivastava et al., 2017) can also happen when different latent samples are all mapped into a single mode in target space.




(a) Quality Discontinuity (b) Proposed Sampling Shift

Figure 1: Red dashed lines denote real data manifolds in pixel space and their inverse images in latent space; yellow circles and squares denote samples for good generation in latent space and pixel space; black circles and squares denote samples for bad generation in latent space and pixel space; the orange dashed lines denote the generator mapping; the blue dashed lines denote sampling shift in latent space; blue shaded and solid blue lines denote the latent distribution and the generated distribution. Due to the continuity of the neural network-based generator, close samples in latent space will be mapped to close images in pixel space, while the images' quality can vary. Cat (a) and Cat (b) are close in pixel space, but Cat (a) loses its right ear. Our method allows bad latent $\mathbf{z}$ to transform to the inverse image of the nearest real manifold, thus avoiding bad generation.




(a) Quality (b) Diversity

Figure 2: Red dashed lines denote real data manifolds; blue shaded and blue lines denote the generated distribution. The generated distributions mapped from a fixed latent distribution reflect the mapping quality. (a) shows the case of bad quality when the generated distribution does not match the real distribution well. (b) shows the case of bad diversity when the mapping misses modes. Our method tries to train a better mapping, as shown on the right of each case.

The above contradiction between the discontinuity in generation data quality (e.g. images) and the continuity of latent distribution as well as mapping function, can be one of the significant factors to the well-known difficulty in GANs training, especially for generators, as one unreasonably hopes to enforce a discontinuous mapping via continuous neural nets. Meanwhile, for functions with continuous nature, the mapping property can also largely affect the generative performance, which is associated with most previous works devoted to improving generative performance.

To tackle the above issues, essentially, we impose an extra (implicit) transform function $z^*(\cdot)$ on the raw sampling $\mathbf{z}$, and the generation can be written by $G(z^*(\mathbf{z}))$. From this perspective, most existing works, including the vanilla GAN, employ $z^*(\cdot)$ with an identical form. Specifically, the implicit function $z^*(\cdot)$ is achieved by updating the latent $\mathbf{z} \sim p(\mathbf{z})$ to minimize $loss_G$ (i.e. $\log(1 - D(G(\mathbf{z})))$) using iterative fast gradient sign method (I-FGSM) (Kurakin et al., 2016) with the parameters of $G$ and $D$ fixed. See Fig. 1 (b) for schematic diagram. On top of the first effort, we propose a novel training strategy named AdvLatGAN to improve the generator mapping $G$ respectively for quality and diversity. See Fig. 2 for schematic diagram. This is achieved by introducing $\mathbf{z}$ iterations (i.e. targeted implicit transforms on $\mathbf{z}$) into GAN training, specifically, updating $\mathbf{z} \sim p(\mathbf{z})$ using I-FGSM to find more optimization-friendly ones in the bi-level optimization of vanilla GAN. Respectively for quality and diversity, we use two iterative updating strategies for $\mathbf{z}$ under different objective functions in training, and we name the two training algorithms respectively as AdvLatGAN-qua and AdvLatGAN-div. The highlights of this paper include:

- We rethink the role of the generator as a continuous function, which may incur quality discontinuity in target space and mode aggregation (i.e. collapse) when the latent samples are directly sampled from a trivial Gaussian distribution. We further propose a composite generation perspective by formulating the mapping from $G(\mathbf{z})$ to $G(z^*(\mathbf{z}))$.

- We propose a new training pipeline for both $G(\cdot)$ and $z^*(\cdot)$, whereby proposed training strategies first train generative mappings respectively for better generation quality and diversity, and then the implicit transform $z^*(\cdot)$ is enforced and computed in an implicit form to obtain an improved distribution for more effective generation. Our training involves adversarial learning under different objective functions for quality and diversity, respectively, leading to new GAN models: AdvLatGAN-qua and AdvLatGAN-div.
- Extensive experimental results on both synthetic and visual data show the promising improvement of our approach on both generation quality and diversity. Our technique is orthogonal to existing other improving techniques.

## 2 Related Work

Implicitly adapting the raw latent vectors $\mathbf{z}$ from Gaussian to improve the generation relates to latent space learning for GAN. While the specific training algorithm using latent vector adapting to mine optimization-friendly samples in training relates to adversarial training, which view the latent adapting in training as an attack to mine hard samples.

**Latent space exploration in GANs.** Techniques have been proposed for latent space exploration for GANs. By utilizing the steerability of GANs in latent space, the work (Jahanian et al., 2019) shifts the distribution to fit camera movement transformation. InterFaceGAN (Shen et al., 2020) decouples entangled semantics with subspace projection to control facial attributes more precisely. In addition to semantics manipulating, some recent works focus on improving latent sampling quality to generate high-quality images. For example, interactive evolutionary computation is used to make the generation process more controllable in (Bontrager et al., 2018), which meanwhile also brings additional manual efforts. To reduce manual intervention, (Roziere et al., 2020; 2021) use Koncept512 (Hosu et al., 2020) as a criterion to improve the quality of fake images. However, as discussed in (Roziere et al., 2020), this strategy is somehow biased, can not work well in each dataset (e.g., Pokemons). The reason is that the quality estimator is not even designated to access the real-data distribution information. Instead, the generation is guided by a predefined score function.

Compared to the above works, we take a relevant idea to (Roziere et al., 2020) that also adapt the latent vectors to an improved condition before input to the generator $G$. Differently, our latent adaption approach incorporates the target distribution.

**Mitigating mode collapse in GANs.** There are also efforts in mitigating the mode collapse, which essentially attributes to the fact that many real-world samples lie in disjoint manifolds (Khayatkhoei et al., 2018). One way is to introduce a collection of generators such that each generator may cover a specific mode since a single generator can hardly fulfill a discontinuous function (Khayatkhoei et al., 2018). However, more works focus on improving the properties of mapping, explicitly obtaining a mapping covering more modes (Elfeki et al., 2019; Meulemeester et al., 2020; Mao et al., 2019). It is worth noting that the generated distribution obtained by this method will contain invalid parts due to the continuity of the latent distribution and the mapping. This calls for a selective sampling of the continuous latent distribution or directly amend the latent distribution, which is rarely noticed in these works.

**Adversarial samples and adversarial training.** Adversarial samples are perturbations that can easily fool deep models, which are indistinguishable to human eyes. Since the seminal work (Szegedy et al., 2013), various adversarial attack methods have been devised (Goodfellow et al., 2014b; Akhtar & Mian, 2018; Kurakin et al., 2016; Madry et al., 2017). To defense the models from attacks, adversarial training is meanwhile developed (Goodfellow et al., 2014b; Wang et al., 2019; Nøkland, 2015). In particular, it is worth noting that several recent studies have shown that adversarial training can bring many benefits to GANs, not just improving robustness. RGAN (Zhang et al., 2020) shows that adversarial training can enhance the generalization of the generator and discriminator. Based on both theoretical analysis and empirical results, the works (Zhong et al., 2020; Liu et al., 2021; Liu & Hsieh, 2019) show that updating discriminator's parameters with the loss calculated by adversarial samples generated by real images can stabilize training and accelerate convergence.

In this paper, we utilize the adversarial sample search method, specifically the iterative fast gradient sign method (Kurakin et al., 2016) to mine optimization-friendly $\mathbf{z}$ samples in the GAN training process, which matches up with the procedure of adversarial training. While most existing works

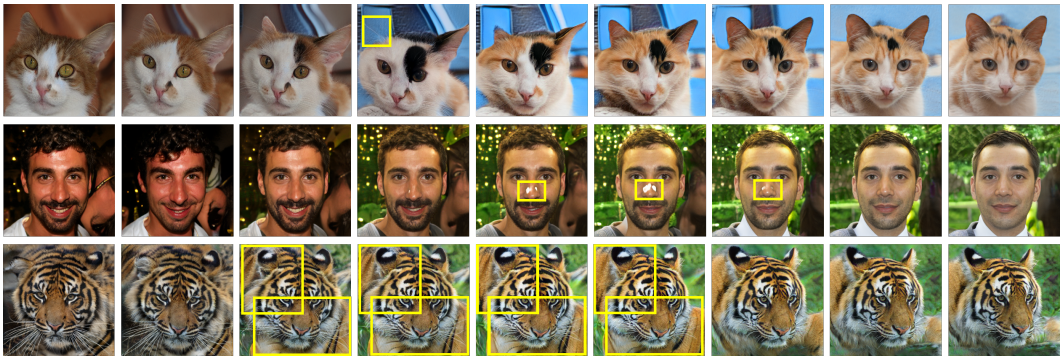


Figure 3: Generation results of the official implementation of StyleGAN2-ada (Karras et al., 2020). When we sample equidistantly from $p(\mathbf{z})$ in the latent space, the sampling in the generated distribution $p_g(\mathbf{x})$ in the pixel space are shown here. The beginning and ending images of each row locate in different manifolds. The intermediate results are sampled by crossing between disconnected manifolds. The discontinuity causes weird generation in the transition, marked with yellow boxes (the first row: the cat misses an ear and its head is crooked; the second row: there exist white spots on the nose; the third row: two bodies share one head).

make perturbations on pixel space (real samples or generated samples), we focus on mining the latent space and making perturbations on latent samples.

## 3 Proposed Method

This section shows how to find the effective transformation $z^*(\cdot)$ and proposes training algorithms AdvLatGAN-qua and AdvLatGAN-div to obtain generation mapping $G$ with better performance respectively on generation quality and diversity. Both methods are derived from the iterative updating of the latent space vectors.

### 3.1 Preliminaries

In the beginning, we first introduce adversarial samples and adversarial training as used in our work to derive $z^*(\mathbf{z})$ and devise training algorithms together with the regularization technique in MSGAN (Mao et al., 2019) for mode seeking.

**Adversarial samples and adversarial training.** Adversarial training in general aims to defend against adversarial attacks. The key idea is that it uses adversarial samples to calculate the optimized loss. In a classifier setting, for each image-label pair $(\mathbf{x}_i, y_i) \in S$ from the labeled training set, the adversarial samples $\mathbf{x}'$ is defined as:

$$\mathbf{x}' = \underset{||\mathbf{x}'-\mathbf{x_i}||_p \leq \epsilon}{\arg\max} \; l(f_\theta(\mathbf{x}'), y_i) \tag{1}$$

Here $\epsilon$ is the radius of the closed ball to make the perturbation small. While $f_\theta$ and $l$ denote the attack target (a neural network) and the corresponding loss (e.g., the cross-entropy loss). $p$ denotes the norm dimension for the perturbation metric.

Given the adversarial samples, adversarial training can be fulfilled by the new loss based on the in total $n$ adversarial samples (together with the raw samples as omitted here):

$$\theta = \min_\theta \frac{1}{n} \sum_{i=1}^{n} l(f_\theta(\mathbf{x}'_i), y_i) \tag{2}$$

**Mode coverage.** MSGAN (mode seeking GAN) (Mao et al., 2019) mitigates mode collapse in conditional generation setting by introducing a regularization term for training $G$ over two latent samples $\mathbf{z}_1$ and $\mathbf{z}_2$ as follows. The hope is that the two generated targets shall be far away from each other as much as possible:

$$\mathcal{L}_{ms} = \frac{d_I\left(G(c, \mathbf{z}_1), G(c, \mathbf{z}_2)\right)}{d_z\left(\mathbf{z}_1, \mathbf{z}_2\right)} \tag{3}$$

where $c$ is the input condition for generation, and $d_I$ and $d_z$ are the distance between two inputs in the target (image) space and latent space, respectively. It encourages the distance to be maintained or even enlarged in the target space to avoid the generated samples being trapped into an aggregation.

By integrating the regularization term with the original objective function of cGANs (Mirza & Osindero, 2014), the final objective for minimization in MSGAN for training $G$ and $D$ is given by:

$$\begin{aligned} \mathcal{L} =& \mathcal{L}_{ori} + \lambda_{ms}\mathcal{L}_{ms} \\ \mathcal{L}_{ori} =& \mathbb{E}_{c,\mathbf{x}}[\log D(c,\mathbf{x})] + \mathbb{E}_{c,\mathbf{z}}[\log(1 - D(c, G(c,\mathbf{z})))] \end{aligned} \tag{4}$$

where $\mathcal{L}_{ori}$ denotes the original objective, $\mathbf{x}$ denotes real data, and $\lambda_{ms}$ is the weight hyper-parameter. Note since Eq. 3 is aimed to be maximized hence one can interpret $\lambda_{ms}$ to be negative.

### 3.2 Latent Space Adversarial Sample Search for Generation Quality

We show how to implicitly transform the latent raw samples from Gaussian to a new distribution via the idea of searching adversarial samples as introduced above. This breaks the dilemma in diversity and quality where the bottleneck is due to the continuous latent distribution together with a continuous mapping function. Moreover, conducting the implicit transform in training to obtain the new (probably disconnected) latent distribution for each discriminator-updating iteration can help more stable and effective training of the network-based generator[1], which cannot map a continuous distribution to a disconnected one with multiple modes.

**Algorithm 1 AdvLatGAN-qua** $t_z, t_D$: the number of steps, other symbols are consistent with vanilla GAN.

**Input:** $p_z$ e.g. Gaussian, $p_r$ e.g. real images distribution; randomly initialized $G$ and $D$;
**Output:** trained generator $G$ and discriminator $D$;

```
while G has not converged do
    // update D:
    for i = 1 to t_D do
        Sample {z_0^(k)}_{k=1}^m ~ p_z;
        // the proposed sampling shift (t-step I-FGSM):
        for i = 1 to t_z do
            Obtain {z_i^(k)}_{k=1}^m by Eq. 5;
        Sample {x_i^(k)}_{k=1}^m ~ p_r;
        Calculate loss_D with {x_i^(k)}_{k=1}^m and {z_t^(k)}_{k=1}^m;
        Update D with loss_D;
    // update G:
    Sample {z^(k)}_{k=1}^m ~ p_z;
    Calculate loss_G with {z^(k)}_{k=1}^m;
    Update G with loss_G;
```

Specifically, we design the following constrained optimization task that encourages the new $\mathbf{z}^*$ to be close enough[2] to $\mathbf{z}$. Its prediction score (as real data) by the discriminator $D$ is as high as possible to improve the generation quality. Recall otherwise the given $\mathbf{z}_0$ may correspond to a poor generation as discussed in Sec. 1. Note here that both $G$ and $D$ are fixed.

$$\mathbf{z}^*(\mathbf{z}_0) = \underset{\mathbf{z}\in\{\mathbf{z}|d(\mathbf{z}_0,\mathbf{z})\leq\epsilon\}}{\arg\min} \log(1 - D(G(\mathbf{z}))) \tag{5}$$

where $\mathbf{z}_0$ denotes the original latent vectors, $\mathbf{z}^*(\mathbf{z}_0)$ denotes the newly computed $\mathbf{z}$, $d(.)$ denotes the distance of two vectors which is an $\ell_\infty$ distance in this paper, in line with the popular protocol in adversarial attack literature (Goodfellow et al., 2014b).

By borrowing the well-developed tools in adversarial attack, the solution of the above problem can be estimated by the so-called iterative fast gradient sign method (I-FGSM) (Kurakin et al., 2016):

$$\mathbf{z}_{i+1} \leftarrow \mathbf{z}_i - \epsilon sgn\{\nabla_{\mathbf{z}_i}\log(1 - D(G(\mathbf{z}_i)))\} \tag{6}$$

From the distribution perspective, the shift can be characterized by:

$$p_z^* = \arg\min_{D(p_z,p_z^0)\leq\epsilon} E_{\mathbf{z}\sim p_z}\left[\log(1 - D(G(\mathbf{z})))\right] \tag{7}$$

where $p_z^0$ denotes the original latent distribution, $p_z^*$ denotes the newly searched distribution, $D(p_z, p_z^0) \leq \epsilon$ means there is a limit to the range of distribution variation. A newly searched $\mathbf{z}$ can be considered as a sample from $p_z^*$ if the sampling distribution is within the perturbation circle. Fig. 4 shows the results by the pre-trained SNGAN model (Miyato et al., 2018) (i.e. the given $G$ and $D$) on STL-10 (Coates et al., 2011).

[1] In literature, it is widely recognized that it is much more challenging to train the generator than discriminator.

[2] We do not want to departure too much from the basic GAN training protocol, and also finally the sampled $\mathbf{z}_0$ itself shall be in good quality for the generation.

Figure 4: Results of latent sample transform on the SNGAN model pre-trained on STL-10. The original generated images are shown on the left, and the images generated with the newly found **z**s after 100 steps iteration of I-FGSM are shown on the right. We find out that the right figures contain richer colors and more realistic details, with the main content remaining consistent. We measure generation quality in a batch by Inception Score, which increases from 6.944 to 7.475.

We develop a novel GAN training algorithm by conducting the above latent distribution shift to the initial sampling of Gaussian in $D$ updating iterations, while the rest of the training algorithm remains consistent with vanilla GAN (Goodfellow et al., 2014a). We present the algorithm in Alg. 1 which we call Adversarial Latent Space Learning for GANs (AdvLatGAN-qua) with a post meaning quality. The latent shift is first computed using the I-FGSM to find the adversarial samples in latent space, then $D$ and $G$ are updated. The iteration continues until enough steps or convergence of $G$.

The above latent shift model does not consider how to avoid mode collapse, as each time only one latent sample is considered for optimization in isolation. Similar constrained optimization formulation can be designed to improve the generation diversity, as shown in the next subsection.

**Algorithm 2 AdvLatGAN-div** $t_z, t_D$: the number of steps, other symbols are consistent with vanilla GAN.

**Input:** $p_z$ e.g. Gaussian, $p_r$ e.g. real images distribution; $\lambda_{ms}$ is a hyper-parameter; randomly initialized $G$ and $D$;
**Output:** trained generator $G$ and discriminator $D$;

**while** $G$ has not converged **do**
  *// update D:*
  **for** $i = 1$ to $t_D$ **do**
    Sample $\{\mathbf{z}_i^{(k)}\}_{k=1}^m \sim p_z$;
    Sample $\{(\mathbf{x}_i^{(k)}, c_i^{(k)})\}_{k=1}^m \sim p_r$;
    Calculate $loss_D$ with $\{(\mathbf{x}_i^{(k)}, c_i^{(k)})\}_{k=1}^m$ and $\{\mathbf{z}_i^{(k)}\}_{k=1}^m$;
    Update $D$ with $loss_D$;
  *// update G:*
  Randomly generate labels $\{c^{(k)}\}_{k=1}^m$;
  Sample $\{\mathbf{z}_0^{(k)}\}_{k=1}^m \sim p_z$;
  *// the proposed sampling shift (t-step I-FGSM):*
  **for** $i = 1$ to $t_z$ **do**
    Obtain $\{\mathbf{z}_i^{(k)}\}_{k=1}^m$ by Eq. 8;
  Calculate $loss_G$ and $loss_{ms}$ (Eq. 3) with $\{c^{(k)}\}_{k=1}^m$, $\{\mathbf{z}_0^{(k)}\}_{k=1}^m$ and $\{\mathbf{z}_t^{(k)}\}_{k=1}^m$;
  $loss \leftarrow loss_G + \lambda_{ms} loss_{ms}$;
  Update $G$ with $loss$;

### 3.3 Latent Space Adversarial Sample Search for Generation Diversity

We take a hard sample mining perspective to improve the mode coverage ability. Given a sample $\mathbf{z}_0$ from the raw latent distribution, we aim to search its paired sample $\mathbf{z}^*$ to form a hard sample pair, in the sense that their generations are close in the target space given $G$. Specifically, we further consider two aspects. First, as hard samples, the two samples shall still belong to the same category $\mathbf{c}$ which can be modeled with conditional GAN. Second, their distance in the latent space shall also be not too close otherwise they may inherently be aggregated to target space due to the nature of continuous mapping. In other words, hard samples causing mode collapse shall be those close-in target space while apart in the latent space, when $G$ is given. Based on the above considerations, we adopt the form of Eq. 3 to design our search procedure as follows:

$$\mathbf{z}^*(\mathbf{z}_0) = \underset{\mathbf{z} \in \{\mathbf{z} | d(\mathbf{z}_0, \mathbf{z}) \leq \epsilon\}}{\arg\min} \frac{d_I(G(\mathbf{c}, \mathbf{z}_0), G(\mathbf{c}, \mathbf{z}))}{d_z(\mathbf{z}_0, \mathbf{z})} \tag{8}$$

The meaning of the symbols and terms are similar to Eq. 3. Akin to AdvLatGAN-qua, we use the I-FGSM to obtain $\mathbf{z}^*$ and we again use $\ell_\infty$ to control the magnitude of the change. Specifically for each update step, we randomly select the first latent vector $\mathbf{z}_0$, and the second vector $\mathbf{z}^*$ is searched by $n$ iterations of the I-FGSM method under the $\ell_\infty$ restriction starting from the neighborhood of $\mathbf{z}_0$, and the iterations are guided by Equation 8.

Figure 5: Results of latent iterations for diversity by using DCGAN (Radford et al., 2016) pre-trained on CIFAR-10 and FACADES, respectively. Fixing the starting point of the latent iterations (the top column), conduct iterations from two directions (the middle and the bottom column). The pair obtained in the anti-diverse direction tend to collapse, while the other shows better diversity.

Table 1 shows the effectiveness of the above paired hard sample seeking approach, whereby the experiments are conducted on CIFAR-10 (Krizhevsky et al., 2009) and Facades (Tyleček & Šára, 2013) datasets. The visual results are shown in Fig. 5. We can see that the pair obtained by solving Eq. 8 tend to collapse (as they are hard samples), while the opposite leads to better diversity by using the inverse form of Eq. 8.

Table 1: Ratio between image and latent distance by using Eq. 8 to derive new $\mathbf{z}^*$. *-inv* means using inverse form of Eq. 8.

| Dataset | Eq. 8 | Eq. 8-inv |
|---|---|---|
| CIFAR-10 | 0.241 | 4.023 |
| FACADES | 0.3785 | 1.3385 |

The above experiments demonstrate the effectiveness of our iteration method, and we further introduce it into the initial training process of MSGAN (Mao et al., 2019). In MSGAN, regularization term 3 is used to regularize the model optimization process, where the selection of the pair is entirely random, without using any heuristic information. We instead use the $\mathbf{z}$ pair that are more inclined to collapse by randomly taking one $\mathbf{z}$ and iterating by the method mentioned above to get another $\mathbf{z}$. We keep the other details of the algorithm unchanged.

Using Eq. 8 to generate adversarial sample pairs to draw the model's attention to address mode collapse, we devise our new diversity enhance GAN algorithm as shown in Alg. 2.

## 4 Experiments

All experiments are performed on a single GPU of GeForce RTX 3090 or GeForce RTX 2080Ti. All datasets we use are publicly available and contain no personal information.

### 4.1 Experimental Setup

We validate the effectiveness of the proposed methods for two parts, respectively for effective latent sampling improvement $z^*$ and better generation mapping $G$. To be explicit, five methods can be derived for different targets:

- 1) Post-training latent sampling improvement fighting quality discontinuity.
- 2) GAN training algorithm for better generation quality using in-training latent sampling improvement: AdvLatGAN-qua.
- 3) GAN training algorithm for generation diversity using in-training latent sampling improvement: AdvLatGAN-div.
- 4) In/Post-training latent sampling improvement for generation quality: AdvLatGAN-qua+.
- 5) In/Post-training latent sampling improvement for generation diversity: AdvLatGAN-div+.

We adopt Inception Score (Salimans et al., 2017) and Fréchet Inception Distance (Heusel et al., 2017) to evaluate the generation quality, which utilizes the classification results of the generated images by the Inception Network to evaluate the performance of the model. A larger value of Inception Score or a smaller value of FID means better performance of the model. Following (Richardson & Weiss,

Table 2: Post-training latent sampling improvement with ResNet and DCGAN architecture on STL-10 (2 initialization per model).

| | | **SNGAN (two initials)** | | | | **WGANGP (two initials)** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | IS(↑) | FID(↓) | IS(↑) | FID(↓) | IS(↑) | FID(↓) | IS(↑) | FID(↓) |
| ResNet | Original | 7.244 | 64.573 | 6.285 | 46.076 | 7.287 | 73.625 | 6.791 | 47.765 |
| | Improved-SPL | **8.116** | **62.184** | **6.833** | **43.099** | **7.743** | **73.443** | **7.369** | **45.121** |
| DCGAN | Original | 6.183 | 46.701 | 7.344 | 64.289 | 6.701 | 48.454 | 7.112 | 69.263 |
| | Improved-SPL | **6.748** | **43.386** | **7.860** | **64.165** | **7.376** | **46.490** | **7.767** | **68.929** |

Table 3: The results of AdvLatGAN-qua and AdvLatGAN-qua+ for CIFAR10.

| Framework | **Inception Score**(↑) | | | **Fréchet Inception Distance**(↓) | | |
|---|---|---|---|---|---|---|
| | vanilla | AdvLatGAN-qua | AdvLatGAN-qua+ | vanilla | AdvLatGAN-qua | AdvLatGAN-qua+ |
| DCGAN | $5.92 \pm 0.05$ | $6.21 \pm 0.07$ | $\mathbf{6.58 \pm 0.34}$ | $46.4 \pm 0.7$ | $41.7 \pm 1.0$ | $\mathbf{40.1 \pm 1.5}$ |
| WGAN | $6.63 \pm 0.05$ | $7.21 \pm 0.04$ | $\mathbf{7.76 \pm 0.07}$ | $32.8 \pm 0.5$ | $27.3 \pm 0.7$ | $\mathbf{27.2 \pm 0.5}$ |
| WGAN-GP | $7.47 \pm 0.09$ | $7.60 \pm 0.06$ | $\mathbf{8.59 \pm 0.10}$ | $24.7 \pm 0.2$ | $22.6 \pm 0.4$ | $\mathbf{18.3 \pm 1.1}$ |
| SNGAN | $7.29 \pm 0.09$ | $7.58 \pm 0.03$ | $\mathbf{8.13 \pm 0.06}$ | $25.5 \pm 0.3$ | $22.3 \pm 0.5$ | $\mathbf{21.9 \pm 0.3}$ |

Table 4: The results of AdvLatGAN-qua and AdvLatGAN-qua+ for STL10

| Framework | **Inception Score**(↑) | | | **Fréchet Inception Distance**(↓) | | |
|---|---|---|---|---|---|---|
| | vanilla | AdvLatGAN-qua | AdvLatGAN-qua+ | vanilla | AdvLatGAN-qua | AdvLatGAN-qua+ |
| DCGAN | $7.18 \pm 0.09$ | $7.33 \pm 0.07$ | $\mathbf{7.79 \pm 0.06}$ | $61.2 \pm 1.2$ | $56.3 \pm 1.0$ | $\mathbf{54.6 \pm 1.3}$ |
| WGAN | $6.51 \pm 0.07$ | $7.62 \pm 0.04$ | $\mathbf{8.16 \pm 0.27}$ | $73.0 \pm 0.2$ | $51.0 \pm 0.6$ | $\mathbf{49.1 \pm 1.2}$ |
| WGAN-GP | $8.86 \pm 0.05$ | $8.90 \pm 0.05$ | $\mathbf{10.32 \pm 0.34}$ | $37.4 \pm 0.4$ | $34.2 \pm 0.9$ | $\mathbf{26.7 \pm 1.1}$ |
| SNGAN | $8.49 \pm 0.09$ | $8.63 \pm 0.08$ | $\mathbf{9.37 \pm 0.05}$ | $36.8 \pm 0.4$ | $34.5 \pm 0.21$ | $\mathbf{30.9 \pm 0.4}$ |

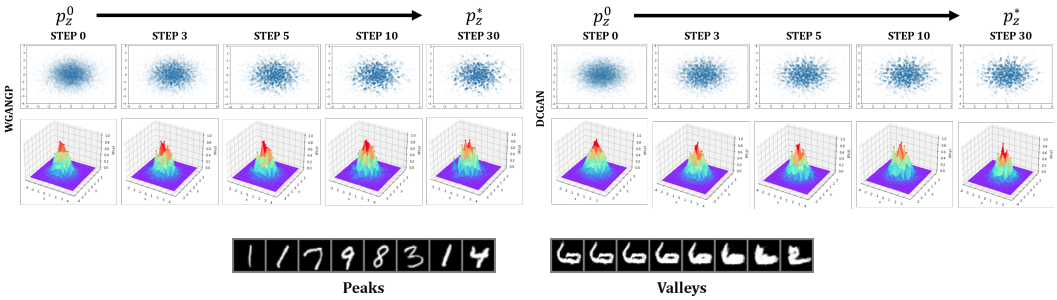



Figure 7: Latent distribution gradually departs from the original Gaussian by conducting the latent transform over its I-FGSM iterations, where two backbones WGAN-GP and DCGAN are used in the experiment. It shows many depressions and peaks on top of the raw Gaussian.

.

2018), we also use JSD to measure the similarity between generated images and real images, which first clusters real images and generated images into the same number of classes using the K-means algorithm, then calculate the similarity between real and fake clusters with respect to the clustering property.

### 4.2 Post-training Latent Sampling Improvement

This section shows the effectiveness of the post-training latent sampling improvement $z^*$ (implicit transform on the original latent sampling from Gaussian), which aims to fight against the quality discontinuity.

**Results on Synthetic Data.** In line with (Metz et al., 2017), we apply the proposed iteration strategy guided by Equation 5 in both Grid and Ring datasets to evaluate its performance. By updating latent space variables, we achieve high-quality generated samples. The result in Fig. 6 shows the effectiveness of our method.

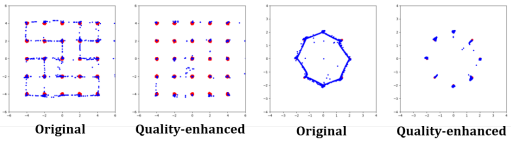



Figure 6: Post-training latent sampling improvement on Grid and Ring by AdvLatGAN-qua.

**Results on MNIST.** MNIST contains 70,000 images of handwritten digits (LeCun et al., 1998). Due to the low complexity of the dataset, the images can be generated with two-dimensional vectors. The two-dimensional latent space probability density distribution can be visualized, granting us better observe the changes in latent distribution during the iterations. We respectively conduct the experiments for the trained DCGAN model and WGAN-GP model, with single-step iteration constraint using $\ell_\infty$ and iterated step size $\epsilon$ is set to 0.03. To better reflect changes in the distribution,

Table 5: FID and JSD results on CIFAR-10.

| Metrics | Models | overall | airplane | automobile | bird | cat | deer |
|---|---|---|---|---|---|---|---|
| FID(↓) | vanilla GAN | 33.102 | 75.710 | 73.279 | 79.712 | 74.941 | 66.849 |
| | MSGAN | 30.225 | 73.083 | **69.518** | 78.258 | 74.525 | 57.778 |
| | AdvLatGAN-div+ | **26.922** | **70.698** | 71.236 | **75.467** | **66.770** | **54.126** |
| JSD(↓) | vanilla GAN | 0.00877 | 0.02811 | 0.02741 | 0.02988 | 0.02753 | 0.02896 |
| | MSGAN | 0.00783 | 0.03004 | 0.02737 | **0.02861** | 0.02571 | 0.02722 |
| | AdvLatGAN-div+ | **0.00533** | **0.02664** | **0.02334** | 0.02869 | **0.02318** | **0.02664** |
| | | dog | frog | horse | ship | truck | |
| FID(↓) | vanilla GAN | 87.178 | 65.967 | 74.619 | 76.311 | 64.851 | |
| | MSGAN | 86.831 | 63.287 | 69.705 | **69.994** | 66.434 | |
| | AdvLatGAN-div+ | **83.932** | **55.281** | **64.764** | 70.616 | **63.506** | |
| JSD(↓) | vanilla GAN | 0.03517 | 0.03686 | 0.04434 | **0.03238** | 0.03032 | |
| | MSGAN | **0.02670** | 0.03381 | 0.04340 | 0.03796 | 0.04093 | |
| | AdvLatGAN-div+ | 0.02700 | **0.03035** | **0.04044** | 0.03469 | **0.02838** | |

we only show the first 30 iterations that can reflect larger changes. The changes of latent distribution and the generated images corresponding to the probability densities at the peaks and valleys of new latent distribution in the WGAN experiment are shown in Fig. 7.

**Results on CIFAR-10 and STL-10.** We equally conduct experiments on CIFAR-10 (Krizhevsky et al., 2009) and STL-10 (Coates et al., 2011). CIFAR-10 is widely used whose complexity is not high, but more diverse and closer to the quality of high-resolution images than MNIST. STL-10 has a higher resolution and richer image representation. We respectively use implicit transformation to the latent space of pre-trained DCGAN and SNGAN, with single-step iteration constraint using $\ell_\infty$ and iteration step size $\epsilon$ set to 0.01. We conduct 100 steps for each iteration process. Inception Score and Fréchet Inception Distance are used to evaluate image quality, and the metrics are calculated over 1500 generated images in STL-10 experiment and 3000 in CIFAR-10 experiment. In both cases, we use two different initializations, each initialized randomly and not specifically chosen. It can be observed that our method has a significant impact on generation quality, with a maximum improvement of 12.03% by Inception Score. The result is shown in Table 2.

### 4.3 Adversarial Latent GAN: Improving the Generation Mapping

This section shows the effectiveness of the GAN training algorithms, which aims to improve the generation mapping $G$ for better performance respectively on quality and diversity.

**AdvLatGAN-qua for Quality Improvement.** We conduct experiments on CIFAR-10 and STL-10, using the mainstream architectures of DCGAN, WGAN, WGAN-GP, SNGAN. We adversarially trained GAN using a quality-enhanced strategy previously mentioned as AdvLatGAN-qua, and the results show that the proposed method can greatly improve the quality of generated images. IS and FID are adopted to evaluate the performance of the model. We conduct only one latent iteration to save computation cost per generator step, and more details will be given in the appendix. We also try to use implicit transformation after training to further improve the generation quality. The result is shown in Table 3 and Table 4. It can be seen that the proposed method has a significant improvement in generative performance compared to that without the latent space transformation strategy. In the WGAN STL-10 setting, IS reaches from 6.51 to 8.16, and FID reaches from 73.0 to 49.1.

**AdvLatGAN for Diversity Improvement.** We conduct experiments on CIFAR-10 for conditional generation and compute FID and JSD, respectively for the entire set of generated images and each class. To improve both the diversity and quality of generated images, we test training results under the iterative diversity improvement using the training algorithm AdvLatGAN-div plus the implicit latent space transformation strategy. We can find that the proposed method is prior over MSGAN reflected by JSD and FID index. The results are shown in Table 5.

## 5 Conclusion

We have shown the important effect of latent space sampling on the quality and diversity of the generation in GANs. We also introduce iterative adversarial training for GANs and develop effective methods, namely AdvLatGAN-qua/div. Extensive experimental results on image data have shown the usefulness of our approach.

## REFERENCES


Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6, 2018.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.

Philip Bontrager, Wending Lin, Julian Togelius, and Sebastian Risi. Deep interactive evolution. In *EvoMUSART*. Springer, 2018.

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

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*. JMLR Workshop and Conference Proceedings, 2011.

Mohamed Elfeki, Camille Couprie, Morgane Riviére, and Mohamed Elhoseiny. Gdpp: Learning diverse generations using determinantal point processes. In *ICML*, 2019.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014a.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

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 *NIPS*, 2017.

Vlad Hosu, Hanhe Lin, Tamas Sziranyi, and Dietmar Saupe. Koniq-10k: An ecologically valid database for deep learning of blind image quality assessment. *IEEE Transactions on Image Processing*, 29, 2020.

Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CVPR*, 2019.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676*, 2020.

Mahyar Khayatkhoei, Ahmed Elgammal, and Maneesh Singh. Disconnected manifold learning for generative adversarial networks. In *NIPS*, 2018.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

Faqiang Liu, Mingkun Xu, Guoqi Li, Jing Pei, Luping Shi, and Rong Zhao. Adversarial symmetric gans: Bridging adversarial samples and adversarial networks. *Neural Networks*, 133, 2021.

Xuanqing Liu and Cho-Jui Hsieh. Rob-gan: Generator, discriminator, and adversarial attacker. In *CVPR*, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *CVPR*, 2019.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *CVPR*, 2017.

Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. In *ICLR*, 2017.

Hannes Meulemeester, Joachim Schreurs, Michael Fanuel, Bart Moor, and Johan Suykens. The bures metric for taming mode collapse in generative adversarial networks. In *CVPR*, 2020.

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

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

Arild Nøkland. Improving back-propagation by adding an adversarial gradient. *arXiv preprint arXiv:1510.04189*, 2015.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*. PMLR, 2017.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*, 2016.

Eitan Richardson and Yair Weiss. On gans and gmms. *arXiv preprint arXiv:1805.12462*, 2018.

Baptiste Roziere, Fabien Teytaud, Vlad Hosu, Hanhe Lin, Jeremy Rapin, Mariia Zameshina, and Olivier Teytaud. Evolgan: Evolutionary generative adversarial networks. In *ACCV*, 2020.

Baptiste Roziere, Nathanaël Carraz Rakotonirina, Vlad Hosu, Andry Rasoanaivo, Hanhe Lin, Camille Couprie, and Olivier Teytaud. Tarsier: Evolving noise injection in super-resolution gans. In *ICPR*. IEEE, 2021.

T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NIPS*, 2017.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020.

Akash Srivastava, Lazar Valkov, Chris Russell, Michael Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *NIPS*, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *Proc. GCPR*, Saarbrucken, Germany, 2013.

Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCV Workshops*, 2018.

Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, volume 1, 2019.

Fangneng Zhan, Hongyuan Zhu, and Shijian Lu. Spatial fusion gan for image synthesis. In *CVPR*, 2019.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*. PMLR, 2019.

Shufei Zhang, Zhuang Qian, Kaizhu Huang, Jimin Xiao, and Yuan He. Robust generative adversarial network. *arXiv preprint arXiv:2004.13344*, 2020.

Jiachen Zhong, Xuanqing Liu, and Cho-Jui Hsieh. Improving the speed and quality of gan by adversarial training. *arXiv preprint arXiv:2008.03364*, 2020.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*, 2017.