



A lightweight ensemble discriminator for Generative Adversarial Networks

Yingtao Xie^a, Tao Lin^{a,*}, Zhi Chen^b, Weijie Xiong^c, Qiqi Ran^b, Chunnan Shang^b

^a College of Computer Science, Sichuan University, Chengdu, China

^b School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu, China

^c University of Electronic Science and Technology of China, Chengdu, China

ARTICLE INFO

Article history:

Received 16 September 2021

Received in revised form 1 May 2022

Accepted 2 May 2022

Available online 11 May 2022

Keywords:

Generative Adversarial Network

Ensemble discriminator

Mode collapse

Instability

Image generation

ABSTRACT

While Generative Adversarial Networks (GANs) have brought immense success in various content-generation tasks, they still face enormous challenges in generating high-quality visually realistic images because of the model collapse or instability during GAN training. One common accepted explanation for the model collapse and instability is that the learning signal provided by the discriminator to the generator become inadequate when the discriminator overconcentrates on the most discriminative difference between real and synthetic images and ignores the less discriminative parts. To this end, we propose a lightweight ensemble discriminator to evaluate the generator from multi-perspective. Borrowing the insights from ensemble learning, several auxiliary discriminators are embedded into one deep model. A novel ensemble loss function is designed to promote the complementarity within the ensemble and train the whole framework in an end-to-end manner. Extensive experiments on datasets of varying resolutions and data sizes prove significant performance improvements over the state-of-the-art GANs. The proposed method can be easily embedded into various GAN frameworks and combined with different loss functions.

© 2022 Elsevier B.V. All rights reserved.

1. Introduction

Over the past several years, Generative Adversarial Networks [1] (GANs) have gained remarkable popularity as they have impressively advanced the state-of-the-art in various content-creation tasks, such as image generation [2–4], image translation [5–7], image inpainting [8,9], and music generation [10]. Generally, a GAN synthesizes realistic data by conducting adversarial competition between two neural networks: a generator and a discriminator. While the generator generates data from random input noise to fool the discriminator, the discriminator distinguishes whether its input data are provided by the generator or from the real data distribution. In such a framework, the discriminator constantly improves its classification ability by learning the difference between the generated and real data. The generator iteratively improves its generation quality based on the learning signal directly provided by the discriminator, and it does not require any manually designed loss function for optimization. This unique adversarial competition provides GANs

with remarkable ability to progressively capture various complex data distribution without *explicitly* defining them. However, despite these advances, it remains a challenge for GANs to produce photorealistic images with high resolution [11–13]. One major cause of this challenge lies in the discriminator network.

In the GAN framework, the discriminator acts as a loss function that provides the generator with an appropriate learning signal to update its parameters, and a more robust discriminator can provide the generator with a better learning signal [14]. In current state-of-the-art GANs, deep Convolutional Neural Networks (CNNs) [2,15] are frequently used to build the discriminator and gradient descent optimization techniques are used to train the networks. A CNN typically comprises several stacked layers of convolution, activation and pooling, designed to extract discriminative features for the recognition task. If there are several features correlated to the task, the gradient descent optimization technique guides the model to choose the most discriminative features and ignore the remaining ones, as the most discriminative features make the steepest descent in the loss functions [16]. Therefore, the discriminator is not incentivized to learn both global and local differences. The generator, on the other hand, only focuses on a subset of the real data distribution. Consequently, the entire GAN framework can be a vulnerable adversarial framework, and instability, divergence, cyclic behavior or mode collapse [17] can easily occur along with GAN training.

* Corresponding author.

E-mail addresses: yingtaoxie@scnu.edu.cn (Y. Xie), scu_lintao@outlook.com (T. Lin), chenzhi@swufe.edu.cn (Z. Chen), 2017200502028@std.uestc.edu.cn (W. Xiong), 77Kiki.Ran@gmail.com (Q. Ran), chunnanshang@gmail.com (C. Shang).

Over the past few years, considerable effort has been devoted to fixing the vulnerability of GANs, and these efforts can be grouped into two directions: architectural modifications and theoretical improvements. In the former direction, researchers proposed various deep networks to support high-fidelity image generation. In particular, novel architectures such as AutoEncoders [18–22], multi-stage CNN [14,23,24], and multi-branch CNN [5,7,25,26] have been used to build better information paths between the discriminator and generator, so that the learning signal can flow to the generator more effectively. As another line of research, researchers have conducted various studies to characterize the convergence properties of the GAN learning procedure theoretically. Based on these theoretical analyses, various normalizations [8,27–29] and regularizations [11,12,30,31] have been designed to optimize the discriminator. Advances in both these directions mitigate the model collapse or instability issues and significantly improve the generation quality. However, state-of-the-art GANs can still generate images that are identical to real images at the most discriminative parts (e.g., texture and color) but very different at the other parts (e.g., shape) [11–13].

Encouraging the discriminator to maintain a more powerful hierarchical representation so that a better learning signal can be provided to update the generator becomes an essential challenge for improving GANs. To address this challenge, we introduce a lightweight deep ensemble and novel ensemble learning loss function to build a discriminator for GANs. Compared with the individual model, an ensemble discriminator comprises several diverse ensemble members that are deliberately designed to complement each other's deficiencies in solving the same task. Therefore, it could provide a more comprehensive evaluation of the generation quality. Despite its potential, constructing an ensemble discriminator for a GAN is challenging, typically because: (1) it raises considerable amount of computational burden to train, evaluate, and deploy multiple independent deep models for one single task; (2) in the context of deep learning, due to sufficient data provided to train each ensemble member, it is generally difficult to maintain appropriate diversity and complementarity within the ensemble [32,33]. This study proposes the construction of multiple discriminators in one single deep CNN model to take advantage of the ensemble discriminator while reducing the computational cost. In the proposed ensemble, each discriminator is forced to maintain a 'plausible' representation at one specific scale and focus on each other's deficiencies. By conducting an end-to-end training, each of the obtained discriminators can evaluate the generated images at one specific scale. Together, they form a powerful ensemble of discriminators that can update the generator better.

The main contributions of this study can be summarized as follows:

- (1) **A unique lightweight deep ensemble architecture.** The proposed ensemble discriminator is a deep ensemble method featuring a unique 'multiple discriminators embedding into one deep model' architecture. Such an architecture provides the following benefits: (i) It significantly reduces the resources required to build an ensemble in the context of deep learning. (ii) It allows the evaluation of the generation quality not only at the final scale (highest resolution) but also at the intermediate scales. Such a structure allows gradients to flow simultaneously at multiple resolutions. As a result, better learning signal is used to update the generator. (iii) Reasonably combining multiple discriminators can generate more gradients; therefore, the vanishing gradient problem when updating the generator can be well handled.
- (2) **A novel ensemble learning loss function.** A novel ensemble learning function is designed to ensure that each ensemble member adaptively focuses on the samples incorrectly classified by the others, thus guaranteeing that the final ensemble will maintain a certain level of complementarity.

- (3) **State of the art performance.** We evaluated the proposed method across several datasets of varying resolutions and data sizes using the state-of-the-art BigGAN [3] and StyleGAN2 [34] models as baselines. Extensive experimental results show that the proposed method provides higher generation quality and also helps defend against mode collapse and instability issues.

The rest of this study is organized as follows. In Section 2, we review related work on improving GANs. In Section 3, the proposed discriminator ensemble is introduced in detail. In Section 4, we describe extensive experiments conducted to demonstrate the general effectiveness of the proposed method. Finally, Section 5 presents conclusions drawn from the experimental results and discusses potential future work.

2. Related work

GAN is still a vulnerable adversarial framework despite its promising performance in various content-generation tasks, and considerable effort has been devoted to designing better GANs. These efforts can be grouped into two directions: theoretical analysis and architectural modification. These two directions are discussed below.

2.1. Theoretical analysis of GANs

As a generative model, the distribution generated by the GAN should be close to the training distribution. However, measuring the "closeness" of the two distributions remains an open question. If the "closeness" between two distributions is incorrectly measured, the gradient can point to a more or less random direction, and then the generator will be given the wrong direction to optimize its parameters [35]. The original GAN uses the Jensen-Shannon divergence as a distance metric; however, this metric leads to the problem of vanishing gradients problems when updating the generator [36]. Many recent studies have focused on introducing new distance metrics into GANs, such as absolute deviation with a margin [22], Kullback-Leibler divergence [2], Wasserstein distance [37], least squares [36,38], and distance on the hypersphere [39]. Researchers have designed various adversarial objectives based on these distance metrics and significantly improved the generation quality. However, according to theoretical analysis and empirical results, each distance metric as the optimization objective has its pros and cons [40]. For instance, although the Wasserstein distance can significantly stabilize the adversarial learning procedure, it also brings non-convergent limit cycles near-equilibrium [36,41].

In addition to various distance metrics, researchers have theoretically analyzed the convergence properties of GAN training and proposed various normalization and regularization techniques. Normalization causes the input features to approach an independent and identical distribution using a shared mean and variance. This property accelerates the convergence of networks and makes deep learning training feasible. Spectral normalization (SN) [29] is the most widely used normalization method for GANs. SN divides the discriminator's weight matrices by their largest singular value. The adaptive normalization (AdaIN) layer was proposed in [27]. AdaIN normalizes the mean and variance of a feature map along its spatial dimensions such that an arbitrary style transfer is processed in real-time. Attentive normalization [8] softly divides the input feature map into several regions based on its internal semantic similarity and then normalizes them separately. Spatially adaptive normalization [28] was proposed for synthesizing photorealistic images, given a semantic input layout. To deal with the model collapse problem in multi-domain image-to-image translation tasks, Shao et al. [6] proposed style regularization to impose constraints on the input noise.

As for regularization, Zhang et al. [11] proposed consistency regularization, widely used in semi-supervised learning, to ensure that the discriminator's output remains unaffected even when its input data has been augmented. DRAGAN [30] penalizes the gradient at the Gaussian perturbations of the training data. Spectral regularization [12] compensates for the spectral distributions of weight matrices to prevent them from collapsing. The method in [31] directly regularizes the squared gradient norm for both the training and generated data. The auxiliary classifier GAN (ACGAN [42]) applies an auxiliary classifier to regularize generated images with label consistency.

Both normalization and regularization play vital roles in neural network training. Specifically, although normalization can accelerate the training process, regularization penalizes the training procedure and prevents neural work from overfitting or collapsing. Both techniques benefit the GAN in creating more plausible content. However, most regularizations in GANs cannot provide overlaid gains when integrated with normalization. A plausible reason for this is that both techniques are motivated by controlling the Lipschitz constant of the discriminator [11,12].

In addition to normalization and regularization, other techniques have also been proposed to optimize the GAN training process. For example, [43] designed a loss function that can adaptively balance the gradients of real and fake images such that the gradients derived from both distributions contribute to the learning process without harming the other. A standard GAN generates realistic data from a predefined prior distribution; however, such a prior distribution may lose semantic information. To deal with this problem, [44] proposed local coordinate coding (LCC) to sample meaningful points from the latent manifold and then produce new data with these codes.

2.2. Architectural modifications on GANs

A GAN can adopt any differentiable network as its components; therefore, it can naturally benefit from the development of deep learning. Researchers in this field have proposed various deep architectures to support high-fidelity image generation. For example, Radford et al. [2] first introduced convolutional neural networks (CNN) into the GANs framework, which is called a deep convolutional GAN (DCGAN). Huang et al. [24] adopted a series of stacked GANs to generate images, from abstract to specific. Borrowing from the style transfer literature, Karras et al. [4] designed a style-based generator for high-fidelity image generation, and the proposed StyleGAN can lead to an automatically learned, unsupervised separation of high-level semantics. An improved version of StyleGAN was proposed in [34]. A multi-scale gradient generative adversarial network (MSG-GAN [14]) modifies the architecture of the discriminator such that the features extracted from various intermediate layers of the generator can be directly fed to the discriminator. Thus, the discriminator can evaluate the generated images at multiple scales.

Despite significant improvements to GAN architecture, training a discriminator remains a process of knowledge distillation, where only the most discriminative features are retained. Thus, the discriminator often focuses on a global structure or local details. Various autoencoders [18–22] have been incorporated into GANs to address this problem. An autoencoder compresses an image into a low-dimensional vector, which is then uncompressed into an image close to the original image. An attractive property of the autoencoder is that it restores the main information missed from the generating process.

Recently, many studies have demonstrated that the generation quality of GANs can benefit significantly from the design of multi-branch discriminators. For example, to address the class imbalance problem in a multi-domain image-to-image translation task,

Zheng et al. [7] divided a strong common discriminator into several discriminators with smaller sizes. To improve the quality of the two-domain image-to-image transfer, Zhou et al. [5] proposed a *one-encoder-dual-decoder* GAN, in which two branches are attached to the discriminator to decode the features and generate images. [25] designed a BSD-GAN, a multi-branch discriminator that can progressively improve the GAN's performance on an unconditional generation task. In BSD-GAN, each branch is used to train the generation ability at a specific scale, and then it is frozen.

The benefits of the multi-branch discriminator are two-fold. *First*, a multi-branch can build a better information path between the discriminator and generator so that the vanishing gradient problem can be better addressed. *Second*, the multi-branch provides candidate features of different scales for the discriminator; therefore, it helps the discriminator maintain a better representation [14]. This study proposes a lightweight ensemble discriminator with similar merits as the existing multi-branch discriminator and also takes a step further. Specifically, a novel ensemble loss function is proposed to *explicitly* force the ensemble members to have a certain level of complementarity within them so that the generator can be evaluated from diverse and complementary perspectives. Moreover, most multi-branch or multi-discriminators are deliberately designed for a specific task or domain (e.g., the work in [14] is purposefully designed for image-to-image translation tasks), and they generally involve major modifications to the existing structure. The method proposed in this study can be easily embedded into various GAN frameworks without significantly changing them, and it can be a general method for improving GAN.

3. A lightweight ensemble discriminator for GAN

We aimed to construct an effective ensemble of discriminators to evaluate the generator from multiple perspectives. Diversity and complementarity are the two pillars of any effective ensemble. To ensure diversity within the ensemble, we embedded multiple discriminators into different intermediate layers of a single deep model so they can be *explicitly* trained with features extracted from different scales. To promote complementarity within the ensemble, we designed an ensemble loss function that enforces ensemble members to compensate for each other's deficiencies. We describe the proposed network structure and loss function in the following sections, where the superscripts r and g denote real and generative distributions, respectively.

3.1. Problem description

The learning process of a GAN involves training generator G and discriminator D simultaneously via an adversarial learning process. Given data $x^r \sim p^r(\mathbf{x})$, the target of G is to approximate the real data distribution p^r over the training data x^r , which is denoted as p^g . G starts with sampling input variable \mathbf{z} from a uniform or normal distribution $p(\mathbf{z})$, and then maps the input variable $\mathbf{z} \sim p(\mathbf{z})$ to new data $G(\mathbf{z}) \sim p^g(G(\mathbf{z}))$. The discriminator D is a classifier that aims to distinguish the real data x^r from the data generated by G . Accordingly, the training target of the GANs can be formulated as follows:

$$\mathcal{L}_G = -\mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} [\log D(G(\mathbf{z}))] \quad (1)$$

$$\begin{aligned} \mathcal{L}_D = & -\mathbb{E}_{x^r \sim p^r(x)} [\log D(x^r)] \\ & - \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} [1 - \log D(G(\mathbf{z}))] \end{aligned} \quad (2)$$

By design, when training the generator G , the loss function in (1) calculates the overall loss on a mini-batch of generated

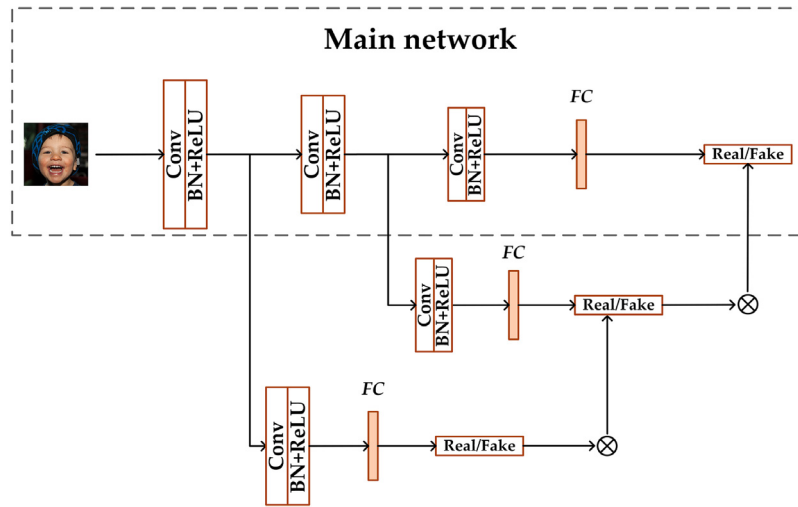


Fig. 1. The architecture of proposed ensemble discriminator. The architecture includes one main network that can be any discriminator structure adopted by the popular GANs, and two auxiliary discriminators that are attached to different layers of the main network. The losses for all the discriminators are computed in a sequential manner, and the loss for 'shallow' discriminator is firstly computed and then used to calculate the modulating factor for the subsequent discriminators. FC: fully connected layer; Conv: Convolutional layer; BN: Batch normalization; ReLU: ReLU activation.

images as the average loss, and the generator minimizes this loss using a gradient descent optimization technique. Meanwhile, the loss function in (2) calculates the average losses on both the real and generated images, and a gradient descent optimization technique is used to minimize such losses for the discriminator. In an ideal case, the loss function in (2) constantly forces the discriminator to improve its ability to distinguish between real and generated images and thereafter provides a better learning signal for the generator to update its parameters. Meanwhile, the loss function in (1) forces the generator to converge to one condition: the generated images converge to the class distribution with the same peak position identical to real data. As such, the generator can be supervised to discover the mapping function from the latent space to the image space and to generate realistic images.

However, given a large-scale dataset of high-resolution images, the model learning guided by the loss function in (1) and (2) is suboptimal. Previous studies have provided strong evidence that real data have a highly multimodal distribution [13,14]. As the training proceeds, the generator cannot benefit much from the generated images, as the discriminator ignores the apparent differences between the generated and real images at the less discriminative parts (e.g., shape) and cannot provide valuable gradients to optimize the generator further. The fundamental scientific problem is how the discriminator can evaluate the generated images from multiple perspectives, as such a better learning signal can be fed back to the generator to enhance the generation quality.

3.2. Network architecture of ensemble discriminator

We attempted to build several discriminators, which are expected to be diverse and complementary to each other, in a single deep architecture. A certain level of diversity within the ensemble ensured that the generator could be evaluated from different perspectives. Complementarity removes redundancy between discriminators and injects more intense competition into the adversarial learning process.

To ensure diversity within the discriminators, we designed a network structure similar to GoogLeNet [45]. GoogLeNet is a classical deep CNN model used for image recognition tasks. In its architecture, GoogLeNet incorporates two auxiliary classifiers into the intermediate layers of its main network. Originally, these

auxiliary classifiers were used to enhance the gradients that get propagated to shallower layers so that they could deal with the vanishing gradient problem. An unexploited merit of this architecture is that the posterior probability provided by these auxiliary classifiers can be *explicitly* used to identify the most complex samples in the same mini batch at different scales. Therefore, the parameters in different layers can be adjusted to complement each other. The starting point of our method lies here.

Fig. 1 shows the structure of the proposed ensemble discriminator. More specifically, we do not change the generator's structure, and the proposed ensemble discriminator comprises two parts: (1) the main network that could be any deep CNN model adopted by popular GANs. (2) Several sub-branches that are connected to different intermediate layers of the main network. As shown in Fig. 1, each sub-branch comprises two parts: (1) several convolutional layers that provide extra capacity for extracting diverse features and (2) an auxiliary classifier that adopts similar structure as the final part of the main network. Because the features captured by the convolutional layers become more fine-grained as the model goes deeper, the proposed structure forces the shallower discriminators (i.e., the discriminators connected to the initial layers of the main network) to be built on low-level features, whereas the deeper discriminators are built on high-level features. In such a multi-branch structure, discriminative features extracted from different scales are used explicitly to construct diverse discriminators, and together, all these discriminators help evaluate the generated samples at different scales. Moreover, as the shallower discriminators are closer to the generator, gradients flow from the discriminator to the generator using a shorter path.

3.3. Ensemble loss function

We propose an ensemble loss function to enforce complementarity within discriminators. Specifically, given a mini-batch of n generated images $\{G(z_i)\}_{i=1}^n$ and a set of M discriminators D_1, D_2, \dots, D_M , where $z_i \sim p(z)$ is the random input noise, $G(z_i)$ is the corresponding generated image, and D_k ($1 \leq k \leq M$) is the k th discriminator in the ensemble (a lower k value indicates D_k is closer to the initial layer of the main network), the loss for the

generator G is the sum of M parts, the k th of which is provided by the k th discriminator and can be calculated as follows:

$$\mathcal{L}_G^k = \frac{1}{n} \sum_{i=1}^n (1 - \overline{p^k(G(z_i))})^2 \times \mathcal{L}_G(G(z_i)) \quad (3)$$

The loss function in Eq. (3) consists of two parts: (i) $\mathcal{L}_G(G(z_i))$, which is a standard generator loss function defined in Eq. (1). (ii) $(1 - \overline{p^k(G(z_i))})$, a modulating factor used to increase the weight of the samples that have been confidently classified as fake data by previous k discriminators. In the modulating factor, $\overline{p^k(G(z_i))}$ is the average confidence assigned by the previous k discriminators to the conclusion that $G(z_i)$ is a real image, and it can be calculated as follows:

$$\overline{p^k_{real}(G(z_i))} = \frac{1}{k} \sum_{j=1}^k p^j_{real}(G(z_i)) \quad (4)$$

where p^j_{real} denotes the estimated probability of the j th discriminator predicting $G(z_i)$ as a real image. p^j_{real} should have a value ranging from $[0,1]$. As most state-of-the-art GANs have removed the softmax layer from their discriminator network, one cannot directly get p^j_{real} from the discriminator. To this end, we first calculate the standard generator loss $\mathcal{L}_G(G(z_i))$ and then calculate p^j_{real} as follows:

$$p^j_{real}(G(z_i)) = \exp(-\mathcal{L}(G(z_i))) \quad (5)$$

$$\mathcal{L}_G = \sum_{k=1}^K \mathcal{L}_G^k \quad (6)$$

After the losses of all discriminators for one mini-batch of generated images are computed, they can be summed to an overall loss using Eq. (6). A high loss value indicates that the generated images are of low quality, and all M discriminators have confirmed this conclusion. This error signal will be back-propagated to the generator with a common gradient descent optimization technique, so that the generator can be optimized to fool all the discriminators in a better way.

The losses for all the M discriminators are calculated sequentially, and the loss for the shallower discriminator (i.e., the one connected to the initial layer of the main network) is computed first. Given a mini-batch of n real images and n generated images, the loss for the k th discriminator can be calculated as follows:

$$\begin{aligned} \mathcal{L}_D^k &= \frac{1}{n} \sum_{i=1}^n (1 - \overline{p^k_{real}(x_i^r)})^2 \times \mathcal{L}(D(x_i^r)) \\ &+ \frac{1}{n} \sum_{i=1}^n (1 - \overline{p^k_{fake}(G(z_i))})^2 \times \mathcal{L}(D(G(z_i))) \end{aligned} \quad (7)$$

The loss function of the discriminators shares the same spirit with that of the generator. In particular, two modulating factors, $(1 - \overline{p^k_{real}(x_i^r)})^2$ and $(1 - \overline{p^k_{fake}(G(z_i))})^2$ have been added to the loss of real image $\mathcal{L}(D(x_i^r))$ and fake image $\mathcal{L}(D(G(z_i)))$, respectively. $\mathcal{L}(D(x_i^r))$ and $\mathcal{L}(D(G(z_i)))$ can be calculated by using the loss function defined in Eq. (2).

As for the modulating factors, $\overline{p^k_{real}(x_i^r)}$ denotes the average probability of a real image x_i^r being predicted as a real image by the previous k discriminators and $\overline{p^k_{fake}(G(z_i))}$ is the average probability of a fake image $G(z_i)$ being predicted as a fake image by the previous discriminators. $\overline{p^k_{real}(x_i^r)}$ and $\overline{p^k_{fake}(G(z_i))}$ can be calculated as follows:

$$\overline{p^k_{real}(x_i^r)} = \frac{1}{k} \sum_{j=1}^k p^j_{real}(x_i^r) \quad (8)$$

$$\overline{p^k_{fake}(G(z_i))} = \frac{1}{k} \sum_{j=1}^k p^j_{fake}(G(z_i)) \quad (9)$$

where $p^j_{real}(x_i^r)$ and $p^j_{fake}(G(z_i))$ are the classification confidence assigned by the j th discriminator on the i th real image x_i^r and fake image $G(z_i)$, respectively. However, as the discriminators do not directly output such confidence in state-of-the-art GANs, we first use Eq. (2) to obtain the discriminators' loss on the real and fake images, e.g., $\mathcal{L}(D(x_i^r))$ and $\mathcal{L}(D(G(z_i)))$, and then calculate $p^j_{real}(x_i^r)$ and $p^j_{fake}(G(z_i))$ as follows:

$$p^j_{real}(x_i^r) = \exp(-\mathcal{L}(D(x_i^r))) \quad (10)$$

$$p^j_{fake}(G(z_i)) = \exp(-\mathcal{L}(D(G(z_i)))) \quad (11)$$

When calculating the loss for the k th discriminator, the modulating factors assign higher weights to the images misclassified by its previous ensemble members, thereby forcing the subsequent discriminators to concentrate on the deficiency of its previous members.

After the losses of all discriminators on the real and generated images are obtained, they can be summed to one loss. The ensemble discriminator minimizes this amount of loss using the gradient descent optimization technique, so that its ability to distinguish real and fake images can be improved. Moreover, as all discriminators are embedded into one deep model, the proposed ensemble discriminator can be trained end-to-end.

The loss functions for the generator and discriminator share the same two-step procedure: we first calculate the losses with the standard GAN loss functions and then calculate the modulating factors accordingly. Thus, as long as all the discriminators adopt the same structure as the main network, we can incorporate various GAN loss functions by simply replacing the standard loss functions in Eqs. (3) and (7) with the target loss function. Moreover, if we replace the loss functions in Eqs. (3) and (7) with those adopted by the conditional-GAN and modify the structure accordingly, the proposed ensemble discriminator can be easily embedded into a conditional-GAN.

3.4. Benefits of the proposed method

The benefits of the proposed method can be derived from the following three aspects: *First*, unlike previous GANs, which evaluate the generated images at one single scale, the proposed method penalizes the generated images, even though they can be perfectly classified with the most discriminative features. As at some scales, the generated and real images show some differences. Therefore, the generated and real distributions have a higher chance of overlapping. One commonly accepted reason for the instability of GAN training is that gradients passing from the discriminator to the generator become inadequate when there is insufficient overlap between real and fake distributions. Theoretically, the proposed method can potentially address instability issues better.

Second, combining several discriminators generates more gradients when updating the generator, which in turn relieves the vanishing gradient problem and builds a shorter information path between the discriminator and generator. This allowed the proposed method to perform a more stable learning process.

Third, unlike the single-model discriminator, which must learn the features highly correlated with the task on every layer, the loss function in (7) forces the subsequent layers and discriminators to focus on the parts that the previous layers and discriminators have incorrectly classified. Therefore, their successors can fix their deficiencies, and together, they provide a stronger classification ability. This in turn improves the generator, as a stronger discriminator can provide a better learning signal to the generator.

4. Experiments

This section describes extensive experiments conducted on several generative tasks to validate the general effectiveness of the proposed ensemble discriminator. The datasets, baseline methods, experimental tasks, and parameter settings are described in the following sections. All codes required to reproduce our work are publicly available at <https://github.com/yingtao-xie/BIGGAN-E>.

4.1. Experimental settings

Datasets. To extensively evaluate the proposed method, we adopted a variety of datasets that exhibit different resolutions and sizes (number of images) as benchmarks, including:

- **CIFAR-100** [46] contains 60k images from 100 classes. In our experiments, each image in CIFAR-100 was scaled to 32×32 resolution and then used to perform ablation studies.
- **CelebA** [47] contains 200k images of celebrity faces. To evaluate the performance of the proposed method on unconditional generation tasks, we scaled each image of CelebA to 128×128 resolution.
- **LSUN** [48] has ten scene categories, and each category consists of approximately 100k natural images of indoor scene. We selected Church, Dining Room and Bedroom categories to train the GAN. More specifically, in our experiment, the images of the Church and Dining Room categories are merged into a new dataset for the conditional generation task, and the images of Bedroom category are used for the unconditional generation task. All images in the LSUN were scaled to 64×64 resolution.
- **CelebA-HQ** [23] and **FFHQ4** contain 30k and 70k images of high resolution (e.g., 1024×1024), respectively. These two datasets were used to evaluate the proposed method's ability to generate high-fidelity images.

Baselines and generative tasks. The major motivation of this study is to propose an ensemble discriminator that can easily be embedded into general GAN frameworks and improve them. To evaluate whether this goal was achieved, two state-of-the-art GAN frameworks, **BigGAN** [3] and **StyleGAN2** [34], were adopted as baselines. We embedded the proposed ensemble discriminator into these two architectures and called them **BigGAN-E** and **StyleGAN2-E**. While we compared BigGAN with BigGAN-E on mid-level resolution datasets, StyleGAN2 and StyleGAN2-E were tested on high-level resolution datasets.

As previous studies [3,12] have pointed out that increasing the batch size or decreasing the network capacity can potentially lead to model collapse and result in generative models with different abilities. We conducted a series of conditional image generation tasks using BigGAN and BigGAN-E with various combinations of batch and channel sizes (the details of the combination are listed in Table 1). BigGAN and BigGAN-E were implemented with similar PyTorch code¹ and trained with the Adam optimizer. To match the previously published work, all BigGAN models were trained with the non-saturating GAN loss. In contrast, the loss function of BigGAN-E was obtained by modifying the non-saturating GAN loss using the procedure described in Section 3.3.

To evaluate the ability of the proposed method to generate high-fidelity images, we conducted a series of unconditional image generation tasks using StyleGAN2 and StyleGAN2-E on high-level-resolution datasets. The implementation of StyleGAN2-E

Table 1

Experiment settings. The experiments are divided into 6 groups: A, B, C, D, E, F. Within each experiment group, the BigGAN and BigGAN-E are adopted. For groups A–F, the batch size inside each group is varied, so that we can study how batch size affects our method. We change the channel sizes between groups to confirm whether the superiority of our method is only achieved under specific discriminator capacity. We adopt a different dataset in group E and F to investigate how different data affect the proposed method. Batch denotes the batch size and CH is the channel size of discriminator.

Setting	Batch	CH	Dataset	Setting	Batch	CH	Dataset
A ₈₋₈	8	8	CelebA	C ₃₂₋₃₂	32	32	CelebA
A ₁₆₋₈	16	8	CelebA	C ₆₄₋₃₂	64	32	CelebA
A ₃₂₋₈	32	8	CelebA	C ₁₂₈₋₃₂	128	32	CelebA
A ₆₄₋₈	64	8	CelebA	C ₂₅₆₋₃₂	256	32	CelebA
A ₁₂₈₋₈	128	8	CelebA	C ₅₁₂₋₃₂	512	32	CelebA
A ₂₅₆₋₈	256	8	CelebA	D ₃₂₋₆₄	32	64	CelebA
A ₅₁₂₋₈	512	8	CelebA	D ₆₄₋₆₄	64	64	CelebA
B ₁₆₋₁₆	16	16	CelebA	D ₁₂₈₋₆₄	128	64	CelebA
B ₃₂₋₁₆	32	16	CelebA	E ₃₂₋₁₆	32	16	LSUN
B ₆₄₋₁₆	64	16	CelebA	E ₆₄₋₁₆	64	16	LSUN
B ₁₂₈₋₁₆	128	16	CelebA	F ₆₄₋₆₄	64	64	LSUN
B ₂₅₆₋₁₆	256	16	CelebA	F ₁₂₈₋₆₄	128	64	LSUN
B ₅₁₂₋₁₆	512	16	CelebA	F ₂₅₆₋₆₄	256	64	LSUN

was based on the PyTorch version of StyleGAN2-ADA [49]. Following the default settings in StyleGAN2-ADA, both StyleGAN2 and StyleGAN2-E models were trained from scratch with RMSprop optimizer. The loss function for StyleGAN2 is the WGAN-GP loss, and the loss function for StyleGAN2-E is obtained by modifying the WGAN-GP loss [50] with the procedure described in Section 3.3.

To further evaluate its performance, we compared our method with several other popular GAN frameworks, including DCGAN [2], WGAN-GP [50], AGE [51], MSG-GAN [14], and LCCGAN++ [44].

Finally, the number of discriminators in the ensemble played a vital role in the proposed method. Intuitively, increasing the number of discriminators improves generation quality, however, this significantly increases the computational burden. Four discriminators were embedded into the different layers of BigGAN-E and StyleGAN2-E to balance the generation quality and computational costs. The detailed structure of BigGAN-E is provided in the Appendix of this paper, and more information about our implementation can be found in our source code.²

Evaluation Metrics. To quantitatively evaluate the proposed method, we adopted Inception Score (IS, higher is better) [52] and Frechet Inception Distance (FID, lower is better) [53] as the evaluation metrics. The FID and IS were computed using 50k synthetic images in all our experiments.

Through the extensive experiments described above, we hope to prove the general effectiveness of the proposed framework when combined with different GAN architectures (e.g., BigGAN and StyleGAN2), generative tasks (e.g., conditional and unconditional image generation tasks), loss functions (e.g., WGAN-GP and non-saturating GAN loss), hyperparameter settings (e.g., different batch sizes and learning rates), and different optimizers (e.g., Adam and RMSprop optimizers).

4.2. Results on mid-level resolution datasets

Previous studies have shown that the performance of a GAN is sensitive to various aspects of its setup, such as the batch size and capacity of the discriminator. Therefore, we must prove the general effectiveness of our proposed method under various settings. We conducted experiments using BigGAN and BigGAN-E for all settings listed in Table 1. All procedures and settings for

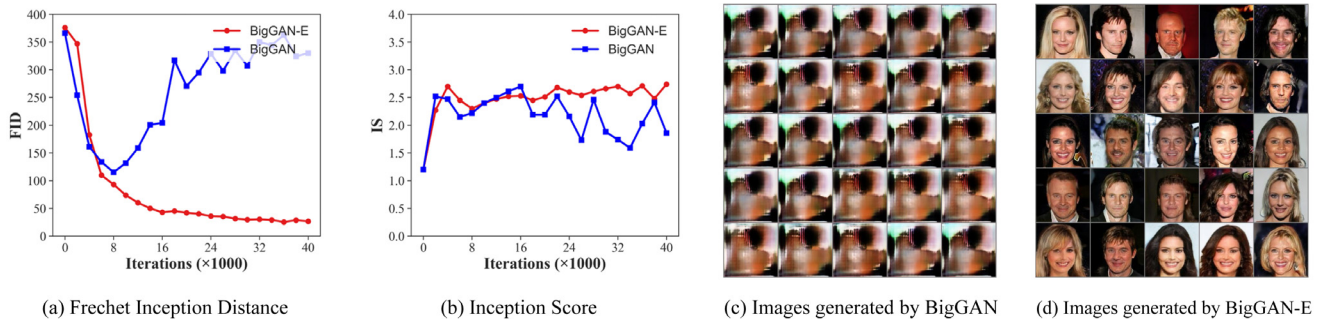
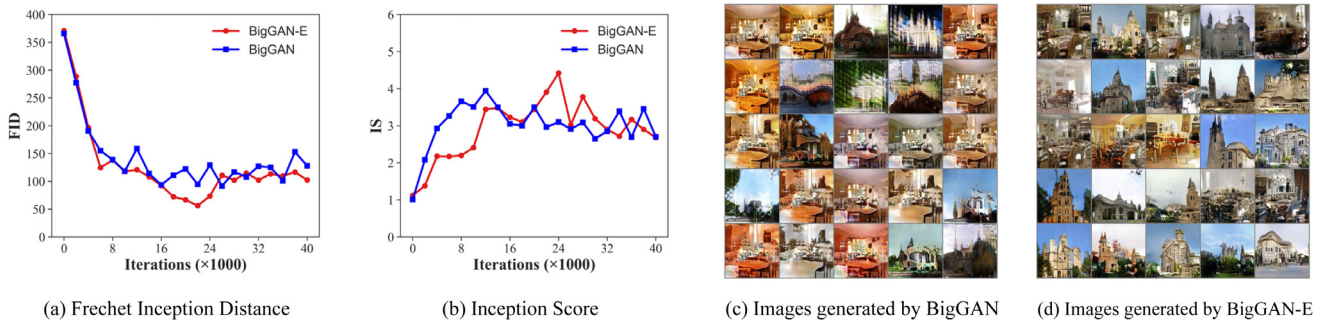
¹ <https://github.com/ajbrock/BigGAN-PyTorch>.

² <https://github.com/yingtao-xie/BIGGAN-E>.

Table 2

Experimental results for different settings. For FID, lower value denotes better performance, while for IS, higher value is better. The best performance for a specific setting is highlighted in red bold-face type. The ● symbol indicates a mode collapse occurs and the metric is the best performance recorded before mode collapse happens.

Setting	FID (↓)		IS (↑)		Setting	FID (↓)		IS (↑)	
	BigGAN	BigGAN-E	BigGAN	BigGAN-E		BigGAN	BigGAN-E	BigGAN	BigGAN-E
A_{8-8}	114.56●	30.21	2.11●	3.09	C_{32-32}	37.27	31.24	2.32	2.85
A_{16-8}	46.82●	33.29	3.12●	3.21	C_{64-32}	56.03●	27.27	2.44●	3.02
A_{32-8}	184.87●	33.49	3.07●	3.17	C_{128-32}	29.76	25.41	2.78	3.04
A_{64-8}	33.36	27.03	2.64	2.72	C_{256-32}	30.24	26.65	2.77	3.23
A_{128-8}	62.44●	24.06	2.76●	2.93	C_{512-32}	28.85	25.91	3.02	3.41
A_{256-8}	115.19●	26.62	2.61●	2.74	D_{32-64}	31.51	25.09	2.79	3.12
A_{512-8}	200.42●	25.15	2.73●	3.22	D_{64-64}	29.45	27.62	2.78	2.73
B_{16-16}	30.64	29.72	2.67	2.87	D_{128-64}	31.01	27.71	2.68	3.12
B_{32-16}	35.79	27.19	2.75	2.77	E_{32-16}	99.41	86.60	3.90	4.10
B_{64-16}	29.28	27.64	2.77	2.83	E_{64-16}	85.62	68.13	3.99	4.37
B_{128-16}	33.52	23.46	2.53	2.69	F_{64-64}	116.28●	89.32	3.29●	3.47
B_{256-16}	35.38	23.07	2.72	2.77	F_{128-64}	85.24	78.05	4.11	4.13
B_{512-16}	33.36	29.36	2.63	2.77	F_{256-64}	91.64	56.19	3.93	4.42

**Fig. 2.** FID, IS and synthetic images of BigGAN and BigGAN-E for the setting A256-8.**Fig. 3.** FID, IS and synthetic images of BigGAN and BigGAN-E for the setting F256-64.

BigGAN and BigGAN-E were identical, except that for BigGAN-E, the discriminator comprised the proposed ensemble.

The FID and IS scores obtained from the experiments are listed in Table 2. For each combination of batch and channel size, the better result is highlighted in red color, and for where mode collapse occurs, the FID and IS are the best scores recorded before mode collapse happens, and all the corresponding metrics are marked by a black dot “●”. From Table 2, it can be easily observed that changing the batch size or channel size results in significantly different performances of BigGAN, suggesting that both the batch size and channel size have severe impacts on BigGAN. Despite this, BigGAN-E outperformed BigGAN in all comparisons can also be observed. In particular, in the setting of A_{512-8} , BigGAN-E improved the FID of BigGAN by 87.45% and IS by 17.95%. On average, BigGAN-E improved the FID of BigGAN and IS by 43.2% and 10%, respectively.

More importantly, although BigGAN represents the state-of-the-art GAN framework that is deliberately designed to defend

the mode collapse problem, mode collapse can still happen to BigGAN. According to Table 2, mode collapse occurred eight times in BigGAN, but none has ever occurred in the proposed method. When the channel size is small, mode collapse has a very high chance of occurring in BigGAN regardless of the batch size, as shown in our group A experiments. Figs. 2(a) and (b) show the changes in the FID and IS scores of setting A_{256-8} . It is observed that after approximately 8000 iterations, the performance of BigGAN starts to decrease and eventually leads to mode collapse. In contrast, as the training proceeded, BigGAN-E steadily improved its performance, and no mode collapse occurred. Figs. 2(c) and (d) show some example images generated by BigGAN and BigGAN-E for setting A_{256-8} . It is clear that mode collapse indeed occurred in the BigGAN.

In addition, when mode collapse does not occur, the proposed ensemble discriminator improves the generation quality. Figs. 3(a) and (b) plot the training history of BigGAN and BigGAN-E for setting F_{256-64} . It was observed that BigGAN converges at the

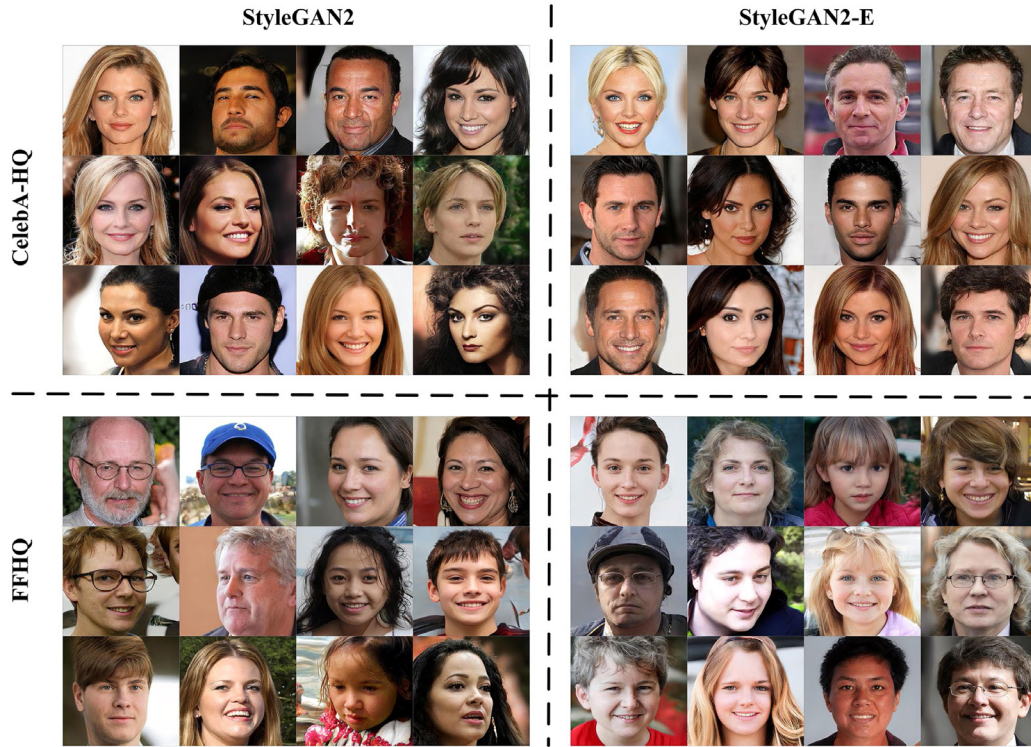


Fig. 4. Random samples generated by StyleGAN2 and the proposed StyleGAN-E on high-resolution (1024×1024) datasets. Please zoom in for better details.

Table 3

Experiments on high-level resolution (1024×1024) datasets. We use author provided scores (denoted by “*”), and otherwise train models with the official code and recommended hyperparameters. The symbol “×” denotes that the corresponding metrics is not available in the original paper.

Dataset	Size	Method	GPU used	Training time	FID (\downarrow)	IS
CelebA-HQ (1024×1024)	30k	ProGAN* [23]	×	×	7.79	×
		StyleGAN* [4]	×	×	5.17	×
		MSG-StyleGAN* [14]	×	×	6.37	×
		StyleGAN2 [34]	1 RTX3090-24 GB	5d 12h 9m	5.34	3.21
		StyleGAN2-E	1 RTX3090-24 GB	6d 8h 24m	5.09	3.56
FFHQ (1024×1024)	70K	ProGAN* [23]	×	×	8.04	×
		StyleGAN* [4]	×	×	4.47	×
		MSG-StyleGAN* [14]	×	×	5.80	×
		StyleGAN2 [34]	1 RTX3090-24 GB	12d 13h 53m	4.64	4.85
		StyleGAN2-E	1 RTX3090-24 GB	14d 12h 31m	4.36	4.97

early stage of training, and there has been no real improvement in its performance since then. In contrast, BigGAN-E constantly improved and eventually converged in the late stage of training (no mode collapse). Example images generated by BigGAN and BigGAN-E for this setting are shown in Figs. 3(c) and (d). Again, it can be observed that the images generated by BigGAN exhibit limitations in terms of variety and quality, whereas the images generated by BigGAN-E have better quality and wider varieties.

All these empirical results demonstrate that the proposed ensemble discriminator can bring significant performance gain to BigGAN and prove that the proposed technique handles the mode collapse problem well.

4.3. Experiments on high-resolution datasets

To evaluate the performance of the proposed method in generating high-fidelity images, we conducted experiments using StyleGAN2 and StyleGAN2-E on *CelebA-HQ* and *FFHQ*. The quantitative results are presented in Table 3.

According to the results in Table 3, compared with StyleGAN2, the proposed StyleGAN2-E increases the GPU training time

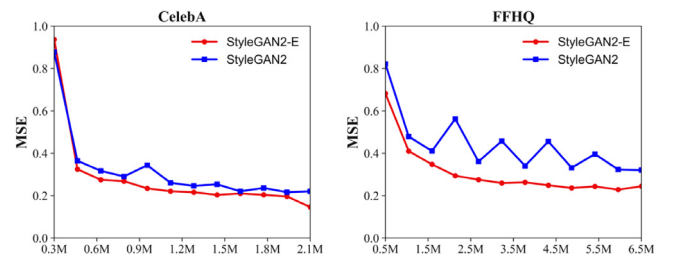


Fig. 5. Training stability of StyleGAN2 and StyleGAN2-E. The figure shows the MSE between images generated from two consecutive checkpoints using the same latent vector (averaged over 16 latent samples). Horizontal axis denotes the number of training images seen by the discriminator.

by approximately 15% and provides better performance metrics. Specifically, StyleGAN2 obtained better FID scores on CelebA-HQ (5.09 vs. 5.34) and FFHQ (4.36 vs. 4.64). The IS scores provided by StyleGAN2-E were also higher than those provided by StyleGAN2 on CelebA-HQ (3.56 vs. 3.21) and FFHQ (4.97 vs.

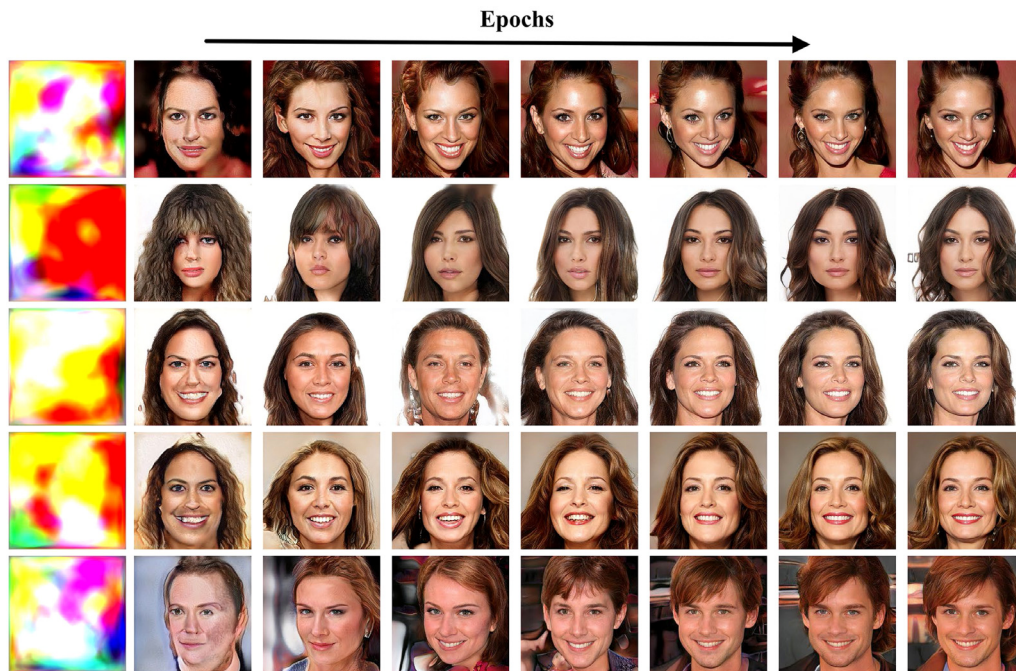


Fig. 6. Visualization on the training process of StyleGAN2-E for some fixed latent vectors. Throughout the whole training procedure, the generator makes steadily improvements to the images generated from fixed latent vectors.

4.85). In addition, to compare StyleGAN2-E with other existing studies, we report the performance of previous studies on the same datasets. According to the results in Table 3, we noted some differences in the author's reported scores. This might be caused by subtle framework differences (e.g., some of the studies implemented their work with Tensorflow while we implemented the proposed method with Pytorch), hardware differences, or variance between runs. Despite the differences described above, StyleGAN2-E provided the best performance in terms of the FID and IS.

Some example images generated by StyleGAN2 and StyleGAN2-E are shown in Fig. 4. Our generated images exhibit fewer phase artifact transitions, which are frequently visible in various GANs.

Stability during training. Stability is an attractive property for any GAN, and it is one of the primary motivations for proposing an ensemble discriminator. To validate whether the proposed method provided better stability, we measured the changes in the generated images for the same fixed latent vector as the training proceeded. The method adopted in [14,54] was used to measure the stability during training, which calculates the mean squared differences between two consecutive images. According to Fig. 5, StyleGAN2 and StyleGAN-E showed the same convergence traits in the early stage of training. However, in the late stage of training, StyleGAN-E converged stably over time, whereas StyleGAN2 continued to vary significantly across epochs.

Methods with high stability enable researchers to obtain a reasonable estimate for the final appearance of images during training, which can help when training procedures take days to weeks. We visualized the changes in the generated images for the same fixed latent vector in Fig. 6. Based on Fig. 6, the stability of StyleGAN2-E can be qualitatively evaluated.

Diversity and complementarity within the ensemble. Diversity and complementarity are two necessary conditions for any ensemble to work. To verify whether the final ensemble exhibits a certain level of diversity and complementarity, we adopted

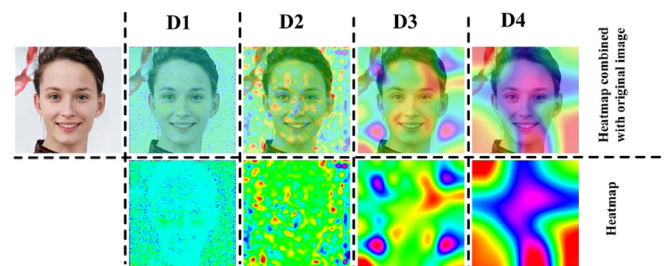


Fig. 7. Visualization of feature maps from different discriminators shows certain level of diversity. Brighter pixels in the second row mean higher activation response.

the visualization technique in [55] to visualize the feature maps of different discriminators in Fig. 7. The second row in Fig. 7 denotes the heatmaps of the discriminators, distinguishing the input data as real or fake. Brighter pixels in the heat maps indicate a higher activation response. The first row in Fig. 7 combines the heatmaps and original image. The four discriminators exhibited a certain level of diversity. In particular, instead of expanding or narrowing the response region of the former discriminators, the subsequent discriminators focused on different parts of the input data. This allows the generation quality to be evaluated from multiple perspectives, and the subsequent discriminators can focus on the deficiencies of their former members.

4.4. Comparisons on LSUN-bedroom

We also conducted experiments on the LSUN-Bedroom to further evaluate the performance of our proposed method against other popular GANs. Specifically, we embedded our ensemble discriminator into SN-GAN [29] (we call the new method as SN-GAN-E) and compared SN-GAN-E with the considered popular

Table 4

Comparisons with SNGAN-E with other popular GANs in terms of IS and FID on the LSUN-bedroom dataset (“*” denotes that the scores are extracted from [44], otherwise, the scores are the mean values averaged from 10 independent experiments).

Methods	LSUN-bedroom	
	FID(↓)	IS(↑)
DCGAN [2]	239.7	2.165
WGAN-GP* [50]	172.6	2.965
AGE* [51]	171.6	2.602
MSG-GAN [14]	145.71	3.021
LCCGAN++ [44]	115.7	4.111
SNGAN-E	94.3	4.727

GANs, including DCGAN [2], WGAN-GP [50], AGE [51], MSG-GAN [14], and LCCGAN++ [44]. The quantitative and qualitative results are presented in Tables 4 and 5.

Table 5 shows that WGAN-GP and DCGAN failed to generate meaningful bedroom images. However, SN-GAN-E, MSG-GAN, and LCCGAN++ can produce images with sharper structures and richer detail. In Table 4, the best performer among the popular GANs is LCCGAN++, with an FID score of 115.7 and an IS score of 4.109. However, the proposed SN-GAN-E outperformed LCCGAN++ by 21.4 and 0.616 when FID and IS were used as the performance metrics.

To further confirm whether the superiority of our method over its competitive counterparts is statistically significant, we supplement the results in Table 4 with statistical tests. In particular, we trained DCGAN, MSG-GAN, LCCGAN++, and our SNGAN-E 10 times on the LSUN-Bedroom, and conducted a Wilcoxon signed-rank test [56] on the obtained IS and FID scores. The Wilcoxon test results are reported in Table 6, confirming that the difference between SNGAN-E and all its competitors is statistically significant. Combined with the fact that the proposed SN-GAN-E provides better IS and FID in Table 4, we conclude that the superiority of SN-GAN-E over popular GANs is statistically significant.

4.5. Ablation studies

This section examines the importance of the key components of the proposed ensemble discriminator by comparing BigGAN-E with its three variants.

(1) **BigGAN**, which adopts only one discriminator without an ensemble.

(2) **BigGAN-half**, which adopts only half the number of discriminators as that in BigGAN-E and trains them with the proposed ensemble loss function.

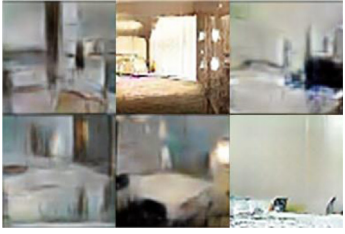
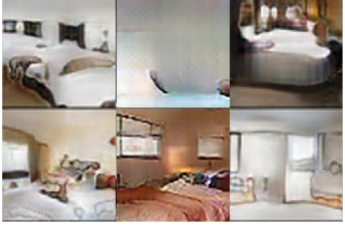
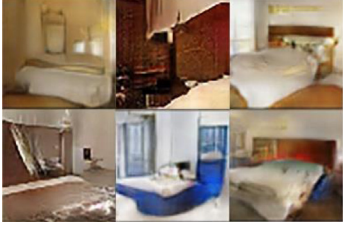
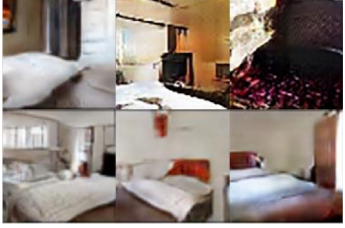
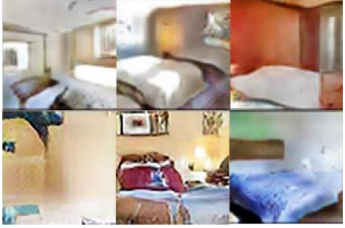

(3) **BigGAN-loss**, which adopts the same number of discriminators as BigGAN-E but trains each discriminator with the standard BigGAN loss function.

CIFAR-100 was adopted as the training set. Inspired by the results in Table 2, we selected the combination of batch and channel size, which made BigGAN achieve its best performance (i.e., B_{256-16}), as the default setting for BigGAN-E and its three variants. Moreover, to check the robustness of the proposed ensemble discriminator toward the learning rate, we trained BigGAN-E and its three variants with different learning rates (0.001, 0.003, and 0.01). Other hyperparameter settings were kept at default settings.

Table 7 presents our experiments with different variants. The best performance for each learning rate is indicated in red. The symbol “●” denotes that mode collapse occurs, and the performance metrics are the best performance recorded before mode collapse happens. Despite the enormous changes in the learning rate, our method provides better performance than its variants.

Table 5

Visual comparisons of SNGAN-E with other popular GANs on LSUN-bedroom.

Methods	LSUN-bedroom
DCGAN [2]	
WGAN-GP [50]	
AGE [51]	
MSG-GAN [14]	
LCCGAN++ [44]	
SNGAN-E	

Robust training schemes are attractive because they indicate how easily a method can be generalized to unseen datasets.

As for the variants, BigGAN-loss exhibited a high tendency toward model collapse. In the settings of three different learning rates, mode collapse occurs for BigGAN-loss. A plausible explanation is that the discriminators in the ensemble exhibit similar behavior without the proposed ensemble loss, and the generator can easily fool all discriminators by generating identical samples. Mode collapse also occurs in BigGAN in one of the three learning

Table 6
Wilcoxon testing result between SNGAN-E and other popular GANs.

Measure	Comparison	R+	R−	p-value	Hypothesis (0.05)
FID	SNGAN-E vs. DCGAN	55.0	0.0	0.005 < 0.05	Rejected
	SNGAN-E vs. MSG-GAN	55.0	0.0	0.005 < 0.05	Rejected
	SNGAN-E vs. LCCGAN++	54.0	1.0	0.007 < 0.05	Rejected
IS	SNGAN-E vs. DCGAN	55.0	0.0	0.005 < 0.05	Rejected
	SNGAN-E vs. MSG-GAN	55.0	0.0	0.005 < 0.05	Rejected
	SNGAN-E vs. LCCGAN++	55.0	0.0	0.005 < 0.05	Rejected

Table 7
Results of ablation studies.

Learning rate	0.0001		0.0003		0.001	
	FID(↓)	IS(↑)	FID(↓)	IS(↑)	FID(↓)	IS(↑)
BigGAN	41.48	6.46	128.38	2.98	138.12●	3.27●
BigGAN-half	34.86	7.01	47.17	6.11	172.77●	4.21●
BigGAN-loss	98.75●	3.65●	85.99●	3.85●	148.59●	3.40●
BigGAN-E	29.62	7.61	35.29	7.05	126.81●	3.15●

rate settings. In addition, the performance of BigGAN was inferior to that of BigGAN-E.

Among the three variants, BigGAN-half performed the best. However, due to the smaller number of discriminators in its structure, BigGAN-half performed worse than BigGAN-E. When the learning rate is set to 0.001 for BigGAN-E, mode collapse occurs in the proposed BigGAN-E, indicating that the proposed method also has limitations.

The ablation of the ensemble loss function and the number of branches in the structure proves that all the key components of the proposed ensemble discriminator contribute toward improving the final performance.

5. Conclusion and future work

Although huge efforts have been made to generate high-resolution photorealistic images, it is still a challenging task to achieve true photo-realism. We argue that the fundamental cause of this challenge lies in discriminator architecture. We developed a novel, simple, effective, and lightweight discriminator ensemble to improve GANs. First, the architecture and loss function of the proposed ensemble discriminator is introduced. Then, we conducted comprehensive experiments to demonstrate that the proposed ensemble discriminator could be easily embedded into popular GAN architectures and improve their performance by effectively defending the mode collapse problem and bringing stability into the training procedure.

Although the general excellence of the proposed method has been effectively demonstrated, there are some limitations to the current method. For example, while the multiple-discriminators-in-one-deep-model architecture significantly reduces the computational cost of building an ensemble in the context of deep learning, currently, the specific architecture of each branch and the number of discriminators in the final ensemble are manually decided. Therefore, the potential of the proposed method cannot be fully explored. In the future, the potential of the proposed ensemble discriminator can be explored in the following ways: (1) combining the proposed method with techniques such as neural architecture search [57] so that the resources can be further reduced, (2) combining it with more advanced loss functions and network structures so that more performance gains can be expected, and (3) applying the proposed method to diverse generation tasks, such as image-to-image translation [6].

CRedit authorship contribution statement

Yingtao Xie: Conceptualization, Methodology, Writing – original draft. **Tao Lin:** Supervision, Writing – review & editing. **Zhi Chen:** Writing – review & editing, Investigation. **Weijie Xiong:** Software. **Qiqi Ran:** Visualization. **Chunnan Shang:** Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Humanities and Social Science Fund of Ministry of Education of China under Grant (17YJCZH210), the Meritocracy Research Funds of China West Normal University (17YC188) and the Key Research and Development Program of Chengdu Science and Technology Bureau under Grant (2019-YF05-02106-GX).

Appendix. Architectural details

Table 8 shows the architecture of ensemble discriminator in BigGAN-E for generating images of 128×128 resolution. The proposed ensemble discriminator consists of four discriminators, three of which are auxiliary discriminators attached to the main network, and the remaining one discriminator is the standard discriminator adopted by the BigGAN. The four discriminators share a common feature extractor of a deep CNN, so the computational resources to build a deep ensemble can be largely reduced. The detailed architecture of the residual block in BigGAN and the architecture of the auxiliary discriminators are given in **Fig. 8**. Specifically, each auxiliary discriminator is comprised of 2 convolutional layers and 1 fully connected layer, and their channel sizes are equal to the layer attached to them.

When generating images of higher resolutions, the number of residual blocks in the main network will be increased, and the BigGAN-E will adjust the layers that attaches the auxiliary discriminators, accordingly.

Table 8

The architecture layout of ensemble discriminator in BigGAN-E for 128×128 images. ch represents the channel width multiplier for each base width (e.g., if $ch = 16$, then $2ch$ in this table represents 32).

Main network	Auxiliary branch
RGB image $x \in \mathbb{R}^{128 \times 128 \times 3}$	
ResBlock down $ch \rightarrow 2ch$	
Non-Local Block (64×64)	
ResBlock down $2ch \rightarrow 4ch$	
ResBlock down $4ch \rightarrow 8ch$	Auxiliary discriminator
ResBlock down $8ch \rightarrow 16ch$	Auxiliary discriminator
ResBlock down $16ch \rightarrow 16ch$	
ReLU, Global sum pooling	
Embed(y)- h + (linear $\rightarrow 1$)	

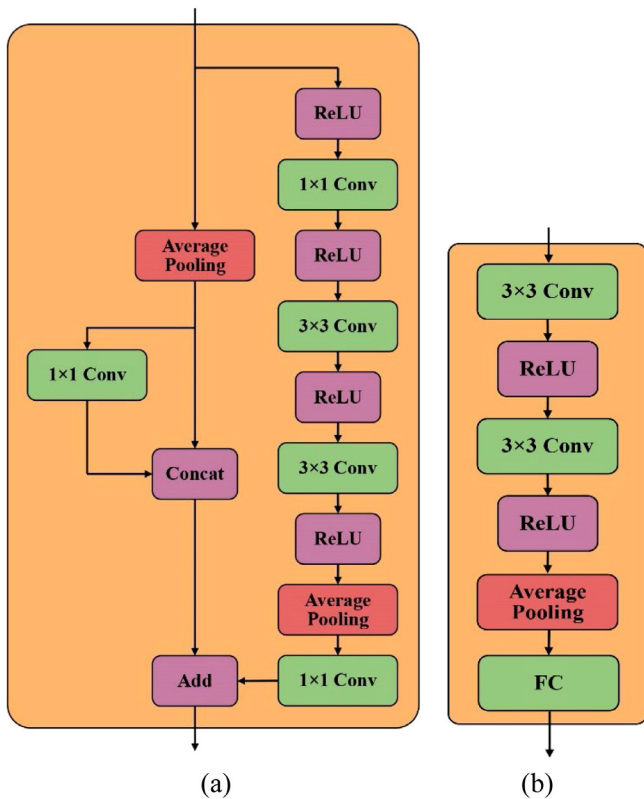


Fig. 8. (a) A typical architectural layout of the residual block for BigGAN's generator. (b) The architectural layout of auxiliary discriminator for the proposed ensemble discriminator.

References

- [1] I. Goodfellow, et al., Generative adversarial nets, in: Neural Information Processing Systems, 2014.
- [2] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, in: International Conference on Learning Representations, 2016.
- [3] A. Brock, J. Donahue, K. Simonyan, Large scale GAN training for high fidelity natural image synthesis, in: International Conference on Learning Representations, 2019.
- [4] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019.
- [5] Y. Zhou, et al., BranchGAN: Unsupervised mutual image-to-image transfer with a single encoder and dual decoders, IEEE Trans. Multimed. 21 (12) (2019) 3136–3149.
- [6] M. Shao, et al., DMDIT: Diverse multi-domain image-to-image translation, Knowl.-Based Syst. 229 (2021) 107311.
- [7] Z.Q. Zheng, et al., Generative adversarial network with multi-branch discriminator for imbalanced cross-species image-to-image translation, Neural Netw. 141 (2021) 355–371.
- [8] Y. Wang, et al., Attentive normalization for conditional image generation, in: Computer Vision and Pattern Recognition, 2020.
- [9] M. Shao, et al., Multi-scale generative adversarial inpainting network based on cross-layer attention transfer mechanism, Knowl.-Based Syst. 196 (2020) 105778.
- [10] H. Zhu, et al., Pop music generation: From melody to multi-style arrangement, Acm Trans. Knowl. Discov. Data 14 (5) (2020) 1–31.
- [11] H. Zhang, et al., Consistency regularization for generative adversarial networks, in: International Conference on Learning Representations, 2020.
- [12] K. Liu, et al., Spectral regularization for combating mode collapse in GANs, in: International Conference on Computer Vision, 2019.
- [13] E. Schönfeld, B. Schiele, A. Khoreva, A U-net based discriminator for generative adversarial networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
- [14] A. Karnewar, O. Wang, MSG-GAN: Multi-scale gradients for generative adversarial networks, in: Computer Vision and Pattern Recognition, 2020.
- [15] J. Ma, et al., DDCGAN: A Dual-discriminator conditional generative adversarial network for multi-resolution image fusion, IEEE Trans. Image Process. 29 (2020) 4980–4995.
- [16] W. Shen, F. Li, R. Liu, Learning to find correlated features by maximizing information flow in convolutional neural networks, in: International Conference on Computer Vision, 2019.
- [17] L. Mescheder, A. Geiger, S. Nowozin, Which training methods for GANs do actually converge, in: International Conference on Machine Learning, 2018.
- [18] Y. Guo, et al., Auto-embedding generative adversarial networks for high resolution image synthesis, IEEE Trans. Multimed. 21 (11) (2019) 2726–2737.
- [19] D. Berthelot, T. Schumm, L. Metz, BEGAN: BOUNDARY equilibrium generative adversarial networks, Arxiv: Learn. (2017).
- [20] M.A. Kiasari, D.S. Moirangthem, M. Lee, Coupled generative adversarial stacked auto-encoder: CoGASA, Neural Netw. 100 (2018) 1–9.
- [21] Z. Yi, et al., DualGAN: Unsupervised dual learning for image-to-image translation, in: International Conference on Computer Vision, 2017.
- [22] J. Zhao, M. Mathieu, Y. LeCun, Energy-based generative adversarial networks, in: International Conference on Learning Representations, 2017.
- [23] T. Karras, et al., Progressive growing of GANs for improved quality, stability, and variation, in: International Conference on Learning Representations, 2018.
- [24] X. Huang, et al., Stacked generative adversarial networks, in: Computer Vision and Pattern Recognition, 2017.
- [25] Z. Yi, et al., BSD-GAN: BRANched generative adversarial network for scale-disentangled representation learning and image synthesis, IEEE Trans. Image Process. 29 (2020) 9073–9083.
- [26] D. Zhu, et al., Diverse sample generation with multi-branch conditional generative adversarial network for remote sensing objects detection, Neurocomputing 381 (2020) 40–51.
- [27] X. Huang, S. Belongie, Arbitrary style transfer in real-time with adaptive instance normalization, in: International Conference on Computer Vision, 2017.
- [28] T. Park, et al., Semantic image synthesis with spatially-adaptive normalization, in: Computer Vision and Pattern Recognition, 2019.
- [29] T. Miyato, et al., Spectral normalization for generative adversarial networks, in: International Conference on Learning Representations, 2018.
- [30] N. Kodali, et al., On convergence and stability of GANs, Arxiv: Artif. Intell. (2018).
- [31] K. Roth, et al., Stabilizing training of generative adversarial networks through regularization, in: Neural Information Processing Systems, 2017.
- [32] J. Zheng, et al., Deep ensemble machine for video classification, IEEE Trans. Neural Netw. Learn. Syst. 30 (2) (2019) 553–565.
- [33] Y. Zhang, et al., Ensemble deep contractive auto-encoders for intelligent fault diagnosis of machines under noisy environment, Knowl. Based Syst. (2020) 196.
- [34] T. Karras, et al., Analyzing and improving the image quality of StyleGAN, in: Computer Vision and Pattern Recognition, 2020.
- [35] M. Arjovsky, L. Bottou, Towards principled methods for training generative adversarial networks, in: International Conference on Learning Representations, 2017.
- [36] X. Mao, et al., On the effectiveness of least squares generative adversarial networks, IEEE Trans. Pattern Anal. Mach. Intell. 41 (12) (2019) 2947–2960.
- [37] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: International Conference on Machine Learning, 2017.
- [38] X. Mao, et al., Least squares generative adversarial networks, in: International Conference on Computer Vision, 2017.
- [39] S.W. Park, J. Kwon, SphereGAN: SPHERE generative adversarial network based on geometric moment matching and its applications, IEEE Trans. Pattern Anal. Mach. Intell. 44 (3) (2022) 1566–1580.

- [40] C. Wang, et al., Evolutionary generative adversarial networks, *IEEE Trans. Evol. Comput.* 23 (6) (2019) 921–934.
- [41] V. Nagarajan, J.Z. Kolter, Gradient descent GAN optimization is locally stable, in: *Neural Information Processing Systems*, 2017.
- [42] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier GANs, in: *International Conference on Machine Learning*, 2017.
- [43] V. Zadorozhnyy, Q. Cheng, Q. Ye, Adaptive weighted discriminator for training generative adversarial networks, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.
- [44] J.Z. Cao, et al., Improving generative adversarial networks with local coordinate coding, *IEEE Trans. Pattern Anal. Mach. Intell.* 44 (1) (2022) 211–227.
- [45] C. Szegedy, et al., Going deeper with convolutions, in: *Computer Vision and Pattern Recognition*, 2015.
- [46] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, Department of Computer Science, University of Toronto, Canada, 2009.
- [47] Z. Liu, et al., Deep learning face attributes in the wild, in: *International Conference on Computer Vision*, 2015.
- [48] F. Yu, et al., Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop, 2015, arXiv preprint [arXiv:1503.03365](https://arxiv.org/abs/1503.03365).
- [49] T. Karras, et al., Training generative adversarial networks with limited data, in: *Neural Information Processing Systems*, 2020.
- [50] I. Gulrajani, et al., Improved training of wasserstein gans, in: *Advances in Neural Information Processing Systems*, 2017.
- [51] D. Ulyanov, A. Vedaldi, V. Lempitsky, It takes (only) two: Adversarial generator-encoder networks, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
- [52] T. Salimans, et al., Improved techniques for training GANs, in: *Neural Information Processing Systems*, 2016.
- [53] M. Heusel, et al., GANS trained by a two time-scale update rule converge to a local Nash equilibrium, in: *Neural Information Processing Systems*, 2017.
- [54] Y. Yazici, et al., The unusual effectiveness of averaging in GAN training, in: *International Conference on Learning Representations*, 2018.
- [55] P.-T. Jiang, et al., LayerCAM: Exploring hierarchical class activation maps for localization, *IEEE Trans. Image Process. (Early Access)* (2021) 15.
- [56] F. Wilcoxon, Individual comparisons by ranking methods, in: *Breakthroughs in Statistics*, Springer, 1992, pp. 196–202.
- [57] Y. Liu, et al., A survey on evolutionary neural architecture search, *IEEE Trans. Neural Netw. Learn. Syst.* (2021) 1–21.