Discriminative Forests Improve Generative Diversity for Generative Adversarial Networks

Anonymous submission

Abstract

Improving the diversity of Artificial Intelligence Generated Content (AIGC) is one of the fundamental problems in the theory of generative models such as generative adversarial networks (GANs). Previous studies have demonstrated that high capacity and robustness of a discriminator is required to achieve high diversity of generated data. However, a high capacity discriminator tends to overfit and guides the generator toward collapsed equilibrium. In this study, we propose a novel discriminative forest instead of the discriminator to improve the capacity and robustness for modeling statistics in real-world data distribution. A discriminative forest is composed of multiple independent discriminators built on bootstrapped data. We proved that a discriminative forest has a generalization error bound, which is determined by the strength of individual discriminators and the correlation among them. Hence, a discriminative forest can own very large capacity without risk of overfitting, which subsequently improves the generative diversity. With the discriminative forest framework, we significantly improved the performance of AutoGAN with a new record FID of 19.27 from 30.71 on STL10 and improved the performance of StyleGAN2-ADA with a new record FID of 6.87 from 9.22 on LSUN-cat.

Introduction

Generating high diversity samples in Artificial Intelligence Generated Content (AIGC) (Cao et al. 2023) inherently requires being able to capture and model complex statistics in real-world data distribution. Generative adversarial networks (GANs) always keep high popularity in the field of AIGC research. As only the discriminator touches real data, the capacity and robustness of discriminator have a critical effect on the generative diversity. Previous studies (Arora, Risteski, and Zhang 2018) showed that the number of modes in the generator's distribution grows linearly with the capacity of the discriminator. Their results imply that higher capacity discriminators are required for better approximation of the target distribution. However, a high capacity discriminator tends to overfit and guides the generator toward collapsed equilibrium (Thanh-Tung, Tran, and Venkatesh 2019). Thus, improving the capacity while keeping robustness of the discriminator is the primary obstacle to ensure that the generator produces data with high diversity.

Many remarkable variations of GAN (Gui et al. 2021) have been proposed to generate more realistic samples. Besides employing the weight normalization and gradient penalty to enhance robustness of discriminator, several recent works investigated the generalization of GANs under multiple adversaries settings. GMAN (Durugkar et al. 2017) is the first endeavor to extend GANs to multiple discriminators. Dropout-GAN (Mordido et al. 2018) integrates adversarial feedback dropout in GMAN, forcing the generator to appease and learn from a dynamic ensemble of discriminators. D2GAN (Nguyen et al. 2017) theoretically analyzed that two discriminators can effectively avoid the mode collapsing problem. PAR-GAN (Chen et al. 2021) extended the D2GAN by using disjoint partitions of input data for multiple discriminators. MCL-GAN (Choi and Han 2022) employs Multiple Choice Learning (MCL) framework to learn multiple discriminators and update the generator via a set of expert discriminators, Albuquerque et al. (2019) offered a new perspective on multiple-discriminator GAN training by framing it in the multi-objective optimization. Generally speaking, GAN with multiple discriminators studies the ensemble strategies of discriminators, including the sampling of training data and the aggregation methods of multiple discriminators. However, for ensemble learning, the Random Forest (RF) (Breiman 2001) has been theoretically and practically shown robustness with a generalization error bound by aggregating a number of randomly built decision trees. Inspired by the RF, if we construct a set of discriminators as a discriminative forest instead of one discriminator to compete with the generator, could this discriminative forest has high capacity and robustness to improve the generative diversity?

In this study, we proposed a *discriminative forest GAN* (Forest-GAN) that consists of a number of discriminators built upon bootstrapping datasets (**Figure 1**). The predictive results of multiple discriminators are aggregated by an aggregation function. Our contributions are in both theoretical proof and the experimental improvement of state-of-the-art on real-world image generation. For the theoretical contributions, (1) we found the global optimality of Forest-GAN approximates to a mixture distribution of bootstrapping datasets. This result indicates that the generator in Forest-GAN has a natural character to defend overfitting of original data. (2) We proved the discriminative forest is robust with an upper bound of generalization determined by the strength of individual discriminators as well as the cor-

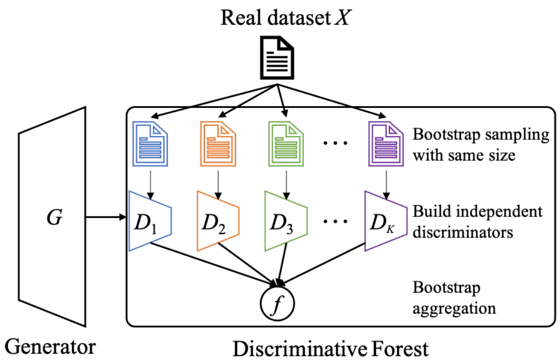


Figure 1: An illustration of the general framework of proposed Forest-GAN.

relation between them, which provides the foundation of understanding that more strength of individual discriminators and less correlation among them leads to a lower generalization error bound. For the experimental contributions, (1) AutoGAN achieves a FID of 19.27 on STL10 (96 × 96, 105K images), significantly improving over the state of the art of 30.71. (2) And StyleGAN2 (ADA) achieves a new record FID of 6.87 from 9.22 on LSUN-cat (256 × 256, 200K images). Besides, Forest-GAN is also flexible as it can be combined with any loss function and weight normalization. Since discriminators are independently trained, Forest-GAN can be deployed with any parallel computing paradigm.

Preliminaries

Generative adversarial networks (GANs)

GANs consist of two neural networks, a *generator* (G) and a *discriminator* (D), which play an adversarial game. Specifically, G learns the distribution p_g over the real data distribution p_r by mapping a random noise vector $z \sim p_z$ to generate a sample $x \sim p_g$. D is trained to maximize the probability of assigning the correct labels to both real examples from training data and fake samples from G. G and D play a mini-max game by optimizing a value function $V(G, D)$:

$$\begin{aligned}\min_G \max_D V(G,D) &= E_{x\sim p_r}[\varphi(D(x))] \\ &\quad + E_{z\sim p_z}[\varphi(1-D(G(z)))] \,,\end{aligned} \tag{1}$$

where the φ is called a *measuring function*. If $\varphi(x) = \log(x)$, Eq. (1) is the objective function used in a classic GAN (Goodfellow et al. 2014). When $\varphi(x) = x$, Eq. (1) becomes a Wasserstein GAN (WGAN) (Arjovsky, Chintala, and Bottou 2017).

Multi-Discriminator GANs

Multi-discriminator GANs consist of one generator and multiple discriminators. D2GAN (Nguyen et al. 2017) has one generator G and two discriminators D_{Θ_1} and D_{Θ_2} that have different parameters Θ_1 and Θ_2, playing three-player mini-max optimization game:

$$\begin{aligned}&\min_G \max_{\{D_{\Theta_k}\}_{k=1}^2} V(G, \{D_{\Theta_k}\}_{k=1}^2) \\ &= \alpha E_{x\sim p_r}[\varphi(D_{\Theta_1}(x))] + E_{z\sim p_z}[\varphi(1-D_{\Theta_1}(G(z)))] \\ &+ E_{x\sim p_r}[\varphi(1-D_{\Theta_2}(x))] + \beta E_{z\sim p_z}[\varphi(D_{\Theta_2}(G(z)))] \,.\end{aligned} \tag{2}$$

Given a sample x, D_{Θ_1} rewards a high score if x is drawn from the data distribution p_r, and gives a low score if generated from the model distribution p_g. In contrast, D_{Θ_2} returns a high score for x generated from p_g whilst giving a low score for a sample drawn from p_r.

PAR-GAN (Chen et al. 2021) extended the D2GAN by using disjoint partitions of input data for multiple different discriminators. More specifically, a training dataset $X = \{x_1, x_2, \cdots, x_n\}$ with a distribution p_X is randomly split into K disjoint partitions $X_1^r, X_2^r, \cdots, X_K^r$, each with the distribution $p_{X_1^r}, p_{X_2^r}, \cdots, p_{X_K^r}$ respectively. Then K different discriminators, $\{D_{\Theta_k}\}_{k=1}^K$, are trained independently on the K partitions. The generator competes with all of these K discriminators. The value function in PAR-GAN is:

$$\begin{aligned}&\min_G \max_{\{D_{\Theta_k}\}_{k=1}^K} V(G, \{D_{\Theta_k}\}_{k=1}^K) \\ &= \frac{1}{K}\sum_{k=1}^K \{E_{x\sim p_{X_k^r}}[\varphi(D_{\Theta_k}(x))] + E_{z\sim p_z}[\varphi(1-D_{\Theta_k}(G(z)))]\} \,.\end{aligned} \tag{3}$$

Compared to PAR-GAN, our proposed Forest-GAN is a general multi-discriminator GAN framework that provides a more effective solution by constructing a number of discriminators upon bootstrapping datasets as a discriminative forest instead of one discriminator. Intuitively, as the number of discriminators increases, the discriminative forest can capture more complex distribution of real data, subsequently leading the generator to produce high quality and diversity of generated samples. However, for PAR-GAN, when the training dataset is limited, more number of discriminators leads to smaller data partitions for each discriminator, makes it more difficult for the discriminators to learn valid distributions on small data partitions. In contrast, Forest-GAN has no such concerns, because its discriminators are constructed on bootstrapping datasets which are sampling repeatedly with replacement from the original dataset. Each bootstrapping dataset has the same size with the original dataset. Taking the advantage of bootstrapping strategy, Forest-GAN theoretically guarantees the generalization bound of discriminative forest, but D2GAN and PAR-GAN have no this properties.

Generalization bound of random forest

RF is an ensemble learning method that applies **b**ootstrap **agg**regat**in**g (bagging) to construct multiple independent decision trees for classification at training time. The predictive output is the class voted by the majority of trees. Given a training set (X, Y) with size of n, an RF works by bootstrapping samples repeatedly K times with replacement from the training set (X, Y) to generate bootstrapping datasets $\{(X_k, Y_k)\}_{k=1}^K$, each with a size of n, and then fitting decision trees $\{f_k\}_{k=1}^K$ on bootstrapping datasets. After training, the classification for unseen samples x' is made by averaging predictions from all individual trees on x': $\hat{f} = \frac{1}{K}\sum_{k=1}^K f_k(x')$ or taking the majority vote.

The generalization error of RF can be bounded as long as the trees in RF are not correlated. Bootstrap sampling is an effective solution to de-correlate trees by showing them different training sets, as sampling with replacement ensures each bootstrap is independent. Since sampling is conducted with replacement, some observations may be repeated in each bootstrapping dataset (X_k, Y_k). For a limited large size n, each dataset (X_k, Y_k) is expected to have a fraction $(1-1/e \approx 63.2\%)$ of unique observations of (X, Y). The remaining observations are called out-of-bag datasets. To decorrelate the trees, RF also includes *feature bootstrapping* since some features are very strong predictors for the response variable. Selecting these features in many trees will cause them to be correlated. In this study, we will adapt the key idea of RF to Forest-GAN, and show its effectiveness in facilitating model generalization and diversity of GANs.

Discriminative Forest GAN

We proposed a novel discriminative Forest GAN (Forest-GAN) that is composed of one generator and one *discriminative forest*. The discriminative forest consists of multiple discriminators constructed upon bootstrap sampling data. Formally, given a real dataset $X = \{x_i\}_{i=1}^n$ that consists of n samples and with distribution of p_{X^r}, we utilized bootstrapping to repeatedly sample K times from X to generate corresponding datasets $\{X_k^r\}_{k=1}^K$, each of size n and distribution of $p_{X_k^r}$. The discriminative forest $D = \{D_{\Theta_k}(X_k^r)\}_{k=1}^K$ is then constructed on bootstrapping datasets, where $D_{\Theta_k}(X_k^r)$ represents the k-th discriminator built on X_k^r with parameters Θ_k, and each Θ_k is initialized independently with same distribution. The value function of Forest-GAN can be defined as Eq (4).

$$\begin{aligned}&\min_G \max_{\{D_{\Theta_k}\}_{k=1}^K} V(G, \{D_{\Theta_k}\}_{k=1}^K) \\ &= E_{x\sim p_{X_k^r}}[\phi(\varphi(D_{\Theta_k}(x)))] + E_{z\sim p_z}[\phi(\varphi(1 - D_{\Theta_k}(G(z))))],\end{aligned} \tag{4}$$

where φ is a measuring function, and ϕ is an aggregating function which can be an average function, max function or majority votes of multiple discriminators. Without loss of generality, we use the average function in the following.

Algorithm 1 shows the training process of Forest-GAN. Intuitively, since only the discriminative forest touches real data in Forest-GAN, "averaging" individual discriminators' outputs does not change the expected value but reduces its variance, subsequently leading to a better and more generalized generator. In the following, we will show Forest-GAN approximates a mixture distribution of bootstrapping datasets. This result indicates that Forest-GAN has a natural character to defend overfitting of training samples. We will then provide the robustness of discriminative forest in Forest-GAN with an upper bound of the generalization error, and show that the bound is determined by the strength of individual discriminators and the correlation between them.

Global optimality

If we take the discriminative forest as a whole, Forest-GAN can be considered as a standard GAN. This indicates that the converge of Forest-GAN can be guaranteed.

We first consider the global optimality of each individual discriminator and the generator. Because each discriminator is trained on its corresponding bootstrapping dataset, for any fixed generator G, the optimal discriminator D_{Θ_k} on bootstrapping dataset X_k^r with distribution of $p_{X_k^r}$ are:

$$D^*_{\Theta_k}(x) = \frac{p_{X_k^r}(x)}{p_{X_k^r}(x) + p_g(x)} . \tag{5}$$

As each bootstrapping dataset is independently sampled from the original dataset with replacement, more than 36% samples of the original dataset are out-of-the-bag. Subsequently, when the training data is limited large size, the bootstrapping dataset distributions $\{p_{X_k^r}\}_{k=1}^K$ are different. According to Eq. (5), the global optimality of discriminators have slightly different.

We next consider the global optimality of generator G. Given K individual discriminators that have achieved their optimal solutions $\{D^*_{\Theta_k}(x)\}_{k=1}^K$. The globally optimized solution of the generator is achieved if and only if G learns the minimum Jensen-Shannon divergence (JSD) of all bootstrapping datasets.

$$\min_G V(G, \{D^*_{\Theta_k}\}_{k=1}^K) \Leftrightarrow \min_{p_g} \sum_{k=1}^K JSD(p_{X_k^r} \parallel p_g) . \tag{6}$$

Algorithm 1: Training process of Forest-GAN.

Input: Real data X^r, with n samples and distribution of p_{X^r}; Random noise z is sampled from a normal distribution p_z; K is the number of individual discriminators in Forest-GAN.
Output: Trained generator G_{Θ^g} and discriminative forest $\{D_{\Theta_k}\}_{k=1}^K$.

Initialize generator G with random weights Θ^g;
for *Number of individual discriminators* K **do**
  Initialize each discriminator D_{Θ_k} with i.i.d weights Θ_k;
  Bootstrap sample from real data X^r with replacement to produce X_k^r, with a sample size of n and distribution of $p_{X_k^r}$;
for *Number of training iterations* **do**
  for *Number of individual discriminators* K **do**
    Sample a batch of m noisy samples $\{z_1, z_2, \cdots, z_m\}$ from a noise prior p_z;
    Sample a batch of m real samples $\{x_1, x_2, \cdots, x_m\}$ from X_k^r dataset;
    Update the discriminator D_{Θ_k} ascendingly by following its stochastic gradient:
    $\nabla_{\Theta_k} \frac{1}{m}\sum_{i=1}^m [\varphi(D_{\Theta_k}(x_i)) + \varphi(1 - D_{\Theta_k}(G(z)))]$.
  for *Number of individual discriminators* K **do**
    Sample a batch of m noisy samples $\{z_1, z_2, \cdots, z_m\}$ from a noise prior p_z;
    Calculate the generator's stochastic gradient competed with D_{Θ_k}:
    $\Delta_{\Theta_k^g} = \nabla_{\Theta^g} \frac{1}{m}\sum_{i=1}^m [\varphi(1 - D_{\Theta_k}(G(z)))]$.
  Update the generator descendingly by following the mean of all generators' stochastic gradients: $\frac{1}{K}\sum_{i=1}^K \Delta_{\Theta_k^g}$.

This result shows the global solution of p_g approximates a mixture distribution of all bootstrapping distributions $\{p_{X_k^r}\}_{k=1}^K$ when the train dataset is limited. This result indicates that Forest-GAN has a natural character to defend overfitting of original samples.

The proof can be found in Appendix A. Although Forest-GAN has similar global optimality in formulation with PAR-GAN, the bootstrapping datasets lead Forest-GAN to learn more general distribution. We also note that a the global optimality of individual discriminators depends on the distribution of $p_{X_k^r}$ and the distribution of $p_g(x)$. But as the condition of Eq. (5), any fixed generator, discriminators approximates to their global optimalities independently. However, learned discriminators are not exactly independent, because during training, the generator receives signals from all discriminators to update itself, and then each discriminator is trained with the generated samples from the shared generator. This issue will increase the correlation among individual discriminators. And we will discuss the effect of correlation of discriminators to the generalization bound of discriminative forest in the next subsection.

Generalization bound of discriminative forest

In this section, we prove discriminative forest can achieve high capacity while keeping robustness with a generalization bound. As the discriminative forest is constructed following the strategy of RF, we utilize the margin function of RF (Breiman 2001) to derive the upper bound of generalization error of discriminative forest. Formally, given K independent decision trees $\{T_{\Theta_k}(x)\}_{k=1}^K$ with the training set (X, Y), the raw margin function of RF is defined as:

$$\begin{aligned} f_{rm}(T_\Theta(X,Y)) =& \frac{1}{K}\sum_{k=1}^{K} I(T_{\Theta_k}(X) = Y) \\ & - \max_{j\neq Y} \frac{1}{K}\sum_{k=1}^{K} I(T_{\Theta_k}(X) = j) , \end{aligned} \tag{7}$$

where $I(\cdot)$ can be an indicator function or activation function that outputs probability ranging [0,1], and $T_\Theta(X, Y)$ is the decision tree with a set of independent weights Θ trained in dataset (X, Y). The margin measures the extent to which the average difference at (X, Y) for the right class exceeds any other class. For a large number of trees, the raw margin function converges to

$$\begin{aligned} f_m(D_\Theta(X,Y)) =& P(T_\Theta(X) = Y) \\ & - \max_{j\neq Y} P(T_\Theta(X) = j). \end{aligned} \tag{8}$$

Specifically in Forest-GAN, the discriminative forest is a binary classifier, where the training set (X, Y) consists of the real data $(X^r, 1)$ and generated data $(X^g, 0)$.

Definition 1. *The margin function of a discriminative forest D with a large number of individual discriminators is defined as :*

$$\psi(D_\Theta(X,Y)) = 2 * P(D_\Theta(X) = Y) - 1 . \tag{9}$$

When the margin function $\psi(D_\Theta(X, Y))$ is lower than 0, the discriminative forest makes a wrong prediction. Thus, the discriminative generalization error can be defined as a probability that D can't distinguish correctly whether an input is real or not.

Definition 2. *The generalization error of the discriminative forest D is defined as*

$$\Psi_D = P(\psi(D_\Theta(X,Y)) < 0) . \tag{10}$$

The larger the margin is, the more confidence the discriminative forest is for classification. Thus, the strength of the discriminative forest can be defined as its expectation of margin function $s = E[\psi(D_\Theta(X, Y))]$, which measures how accurate each individual discriminator is. We define the correlation between two members of the discriminative forest as $\rho(D_\Theta, D_{\Theta'})$, and the mean value of correlation is $\bar{\rho} = E[\rho(D_\Theta, D_{\Theta'})]$, which is a result of the same architecture and feature space. The upper bound of discriminative error can be derived in terms of the expected strength s and mean correlation $\bar{\rho}$.

Theorem 1. *The upper bound on the generalization error of the discriminative forest D is given by*

$$\Psi_D \leq \bar{\rho}(1 - s^2)/s^2. \tag{11}$$

The proof can be found in Appendix B. **Theorem 1** indicates that the generalization error of discriminative forest is dependent on two terms, namely the strength s of individual discriminators and the mean correlation $\bar{\rho}$ among them in terms of the margin function. Hence, these two terms, s and $\bar{\rho}$, form the foundation of understanding how to reduce the upper bound of generalization error. The more strength of individual discriminators and less correlation among them, the lower is the generalization error bound of the discriminative forest. But the limitation of **Theorem 1** is that it doesn't give a way to calculate the correlation and strength. Specifically, sharing of generator among multiple discriminators could make them correlated. We will discuss some strategies to reduce the correlation in *Discussion* section.

Results

Experimental settings. We conducted evaluation on simulated data, real-word images STL10 (96×96, 105K images) (Coates, Ng, and Lee 2011) and LSUN-Cat(256×256, 200K images) (Yu et al. 2015), respectively. We first investigated the effectiveness of Forest-GAN and visualized the changes of diversity in generated samples on simulated data. The simulated data was generated using a 2D Gaussian Mixture Model (2D-GMM) with 9 mixture components. Then we employed the AutoGAN (Gong et al. 2019) as a basis model on STL10 dataset to evaluate the effect of capacity of discriminative forest by controlling the number of total parameters. We also employed StyleGAN2 (Karras et al. 2019) as the basis model on LSUN-cat dataset to evaluate the improvement of Forest-GAN on the large and high-resolution dataset. Since more efficient generalization error bound may be established when the number of individual discriminators K is larger, we evaluated Forest-GAN with K=1,2,5,10,20,50 on simulated data. But considering the time-consuming of training, we evaluated Forest-GAN with K=1,2,5,10 on STL10 and with K=1,5 on LSUN-cat. To

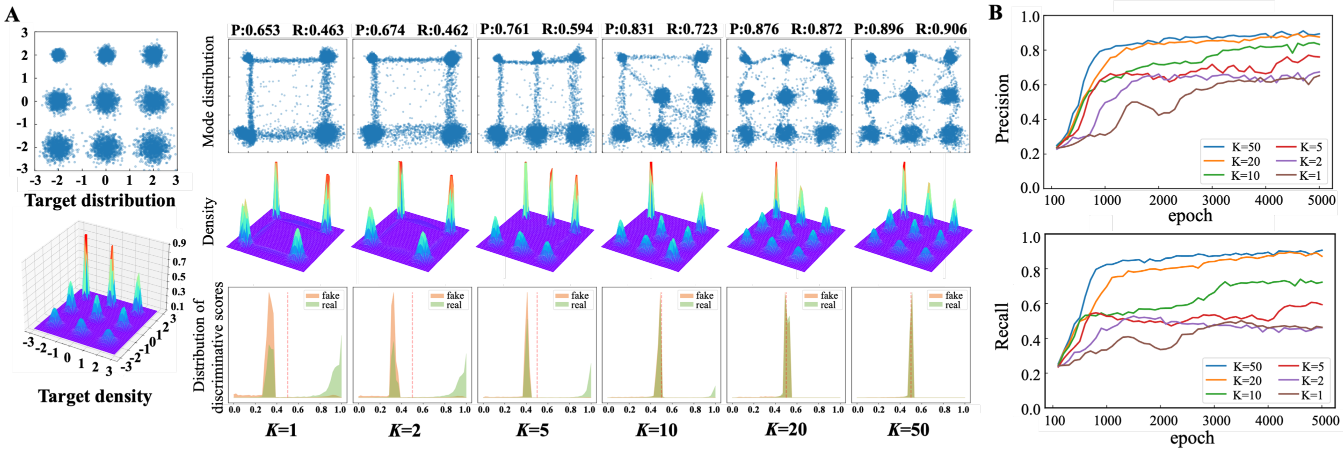


Figure 2: Illustration of effectiveness of discriminative forest on simulated data. (A) Comparison of mode distribution, density estimation and distribution of discriminative forest's scores. (B) Precision and recall curves of Forest-GAN during training.

highlight the effectiveness of Forest-GAN, we kept the backbone of basis methods, other than the number of discriminators. Implementation details are described in Appendix C.

Metrics. As common metrics such as Fréchet Inception Distance (FID) (Heusel et al. 2017) and Inception Score (IS) (Salimans et al. 2016) are only able to distinguish sample quality but the coverage of the sample distribution, we also evaluated the generated distributions of GANs using precision (**P**) and recall (**R**) (Sajjadi et al. 2018; Kynkäänniemi et al. 2019), which disentangle the divergence into two dimensions to separately measure the generative quality and diversity. The higher score of Precision means the model can learn more realistic modes from real dataset, and the higher score of Recall means more modes are recalled from real dataset. While the original formulation of precision and recall fails to consider distribution density in the high-dimensional feature space, we modified the conditional function by adding a density constraint $\lambda \geq \frac{d_g}{d_r} \geq \frac{1}{\lambda}$, where d_g and d_r are the radius of hypersphere in generated space and real data space, and λ was optimized as 3 on simulated data. Thus, we set $\lambda = 3$ in this study. Above metrics are calculated based on the high-dimensional feature space of the Inception-V3 network (Simonyan and Zisserman 2014).

Effectiveness of discriminative forest on simulated data

We first used simulated data to investigate the capability of Forest-GAN in improving generative diversity and enhancing density estimation. The simulated dataset includes 10,000 points with 9 mixture components, where the centroids of these components are [-2, 2], [0, 2], [2, 2], [-2, 0], [0, 0], [2, 0], [-2, -2], [0, -2], [2, -2] and variances are 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, respectively. As each component has the same number of points, the densities of 9 components are decreasing. Capturing accurate density is more challengeable than only learning the components.

Improving generative diversity. We evaluated generative diversity by qualitatively visualizing the generated mode coverage and quantitatively calculating precision and recall. The ground truth and generated distributions produced by Forest-GAN are shown in **Figure 2A**. When $K = 1$, the learned distribution only has four components out of 9 components in the real distribution. More components are learned as K increases. Forest-GANs learn all 9 modes until $K \geq 20$, and its performance is further improved when K is 50, achieving a precision of 0.896 and recall of 0.906.

Enhancing density estimation. Comparing to mode coverage, density estimation is to learn the probability mass function, requiring more close approximation to real distribution. We calculated the density of each component using 2D-GMM, and visualized the densities of simulated and generated modes in 3D plots (**Figure 2A**). As the number K increases, the learned densities of components are significantly improved. Both Forest-GAN with $K = 20$ and $K = 50$ learn all 9 components, but Forest-GAN at $K = 50$ provide more accurate density estimation of the target distribution. These results indicate that a larger K provides more promising generative capability.

Reducing overfitting of discriminative forest. When the number of discriminators is small, Forest-GANs suffer from the mode collapse. To investigate the reason, we visualized the distribution of discriminative forest's outputs (**Figure 2A**). It's obvious that the discriminative forest is overfitting when $K = 1$, and the overfitting is inhibited as the K increases. We can draw a similar conclusion according to the precision and recall curves during training (**Figure 2B**). The recall of Forest-GANs with $K = \{1, 2\}$ decreases after training 1,000 epochs, which indicates that models with small K are prone to overfit, guiding the generator towards mode collapse. As K increases, discriminative forest inhibits overfitting and leads the generator to be more diverse.

Enhancement of AutoGAN by discriminative forest on STL10 dataset

We further evaluated the enhancement of discriminative forest framework on existing GANs. We first employed AutoGAN (Gong et al. 2019) as the basis model on a popular real-world image dataset, STL10 (96×96, 105K images).

Table 1: The performance of AutoGAN is enhanced by discriminative forest on STL10 in different parameter settings. Para indicates the total number of parameters in discriminator forest. 1M means 1 Million.

Method	Para=1M				Para=2M				Para=5M				Para=10M			
	K=1	K=2	K=5	K=10	K=1	K=2	K=5	K=10	K=1	K=2	K=5	K=10	K=1	K=2	K=5	K=10
FID ↓	30.71	28.01	25.88	24.52	29.10	25.46	23.08	22.72	27.96	26.98	22.21	19.82	31.23	25.79	21.78	19.27
IS ↑	9.42	9.82	9.82	9.64	9.23	9.66	10.09	10.07	9.30	9.53	9.79	10.03	9.19	9.56	8.89	10.12
Precision ↑	0.538	0.542	0.564	0.556	0.538	0.573	0.577	0.565	0.530	0.600	0.593	0.594	0.515	0.578	0.606	0.607
Recall ↑	0.471	0.509	0.540	0.554	0.520	0.556	0.562	0.572	0.535	0.574	0.574	0.605	0.501	0.548	0.562	0.596

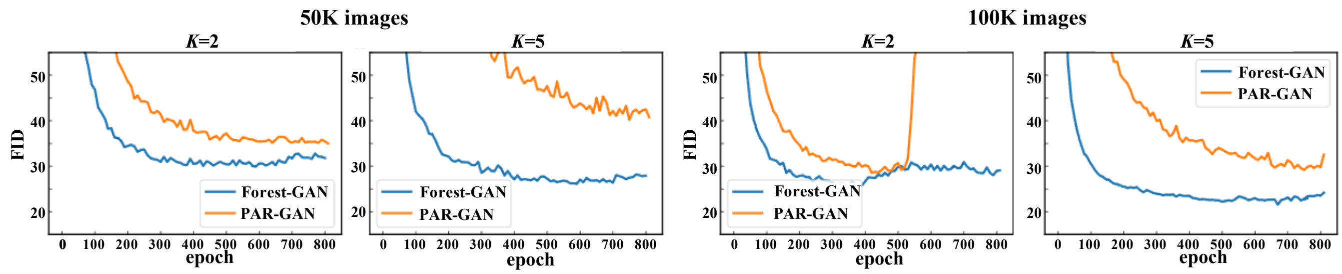


Figure 3: Performance comparison between Forest-GAN and PAR-GAN on STL dataset with 100K samples and 50K samples.

The details of AutoGAN are shown in Appendix D.

Keeping same architecture of individual discriminators. We kept the backbone and training hyperparameters of the generator and discriminator in AutoGAN, and only increased the number of discriminators according to the discriminative forest framework settings. All discriminators have the same architecture, but are independently initialized and trained on their own bootstrapping datasets. We investigated the performance of Forest-GANs with K=1,2,5,10. The discriminator of the original AutoGAN has 1 Million parameters. For K discriminators in a discriminator forest, the total parameters become as K times, as all discriminators have the same architecture. Thus, with keeping same architecture of discriminators, we sought to compare the results of Para=1M (K=1), Para=2M (K=2), Para=5M (K=5) and Para=10M (K=10) in **Table 1**. These results show that when K increases, discriminator forest indeed improves the generated sample quality of AutoGAN's generator in terms of FID, IS and Precision, and generative diversity in terms of Recall. The Forest-GAN with K=10 significantly improved the generated sample quality of AutoGAN with a new record FID of 19.27 from the FID of 30.71 achieved by original AutoGAN, as well as a generative diversity with recall of 0.596. As training more epochs (**Figure D2**), we observed that the model with fewer discriminators tends to deteriorate after it reaches an optimal value, but the discriminator forest with more discriminators can effectively alleviate such dilemma.

Keeping total parameters in discriminator forest. For a fair comparison, we also evaluated the effectiveness of Forest-GAN with the same total parameters. The number of parameters in discriminator forest is controlled by scaling the width in each layer rather than adjusting its depth. Same as above comparison, we set up four experiments with total parameters of 1M, 2M, 5M and 10M, respectively. Each of them conducted four experiments with K=1,2,5,10, which indicates that there are Para$/K$ parameters per discriminator (**Table 1** and **Figure D2**). With the same number of parameters, in other words, discriminator forest has the same capacity, the generated sample quality is improved as the K increases in terms of FID, IS and Precision, as well as the generative diversity in terms of Recall. However, with the same number of discriminators, when the total amount of parameters increases, in other words, the discriminator forest has more capacity, but the generative quality and diversity are not improved. These results indicate discriminator forest can own very large capacity without worries about overfitting so that capturing and modeling more complex statistics in real-world data distribution.

Compared with PAR-GAN. Considering the SOTA performance of PAR-GAN and its correlation with Forest-GAN, we compared them on STL10 dataset with 50K and 100K images (**Figure 3**). It is obvious that Forest-GAN outperforms PAR-GAN under all comparisons. When compared on the STL10(50K images), Forest-GAN becomes slightly better as the number of discriminators K increases. However, PAR-GAN becomes worse. When compared on the STL10(100K images), PAR-GAN(K=2) suffers from mode collapse. Although PAR-GAN(K=5) improves the performance, it's still significantly worse than Forest-GAN. These results are not surprised. PAR-GAN builds multiple discriminators on disjoint partitions of input data. Subsequently, PAR-GAN could suffer from underfitting, especially when it has large number of discriminators trained on a small dataset, as the partitions of the small dataset have fewer samples that are not enough to optimize individual discriminators. While Forest-GAN builds multiple discriminators on bootstrapping datasets that have the same size with the original dataset. Thus, Forest-GAN has no such issue.

Improvement on the high-resolution images

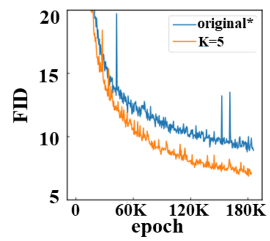
To illustrate the validity of Forest-GAN on more complex real-world images, we selected StyleGAN2-ADA (Karras

A

The improvement of StyleGAN2-ADA by discriminative forest framework.

	original	K=5
FID	9.22	**6.87**
Precision	0.595	**0.625**
Recall	0.202	**0.279**
PPL↓	543	**485**

B



C

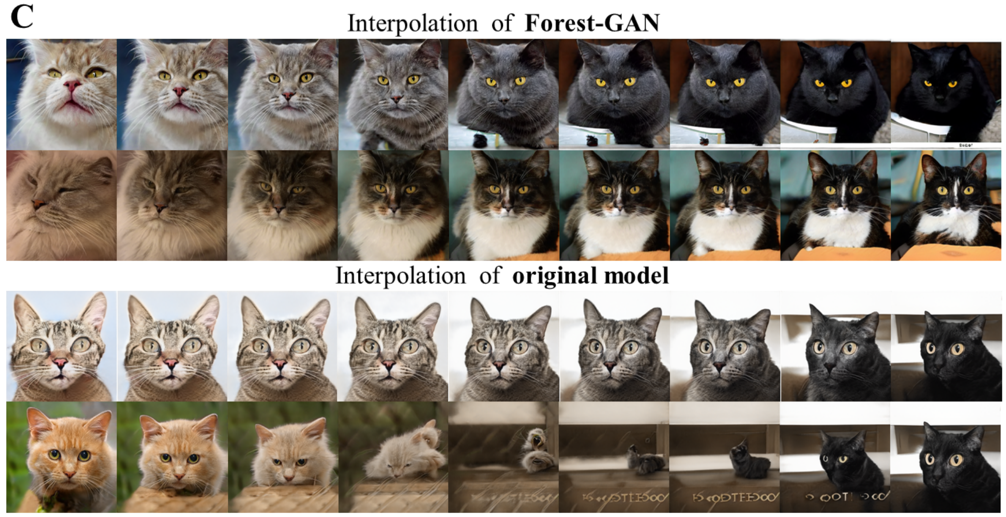


Figure 4: Performance comparison of StyleGAN2-ADA and Forest-GAN. (A) The improvement by discriminative forest. (B) The FID curves during training process. The FID curve marked with original* is reproduced by running original StyleGAN2-ADA. (C) The style space interpolation.

et al. 2020) as the basis model on a large and high-resolution image dataset, LSUN-Cat (256 × 256, 200K images). The raw images were prepared same as StyleGAN2-ADA. We fitted StyleGAN2-ADA into the discriminative framework with 5 discriminators, while keeping others settings.

Improving the quality and diversity. Figure 4A&B shows the improvement of StyleGAN2-ADA by discriminative forest. The discriminative forest with K=5 improved the performance of StyleGAN2-ADA with a new record FID of 6.87 from 9.22, even higher than that trained on 1.6M images, whose best FID is 6.93 (Karras et al. 2019). The generative diversity is also significantly improved in terms of Recall. Besides, according to the FID curves in **Figure 4B**, discriminator forest can accelerate the optimization of the generator. Note that the FID curve of original* is reproduced to compare the convergence of training. Both two models may be further improved if training more epochs.

Better interpolation generation. Since above results show the generative diversity is significantly improved, the style space interpolation is further explored. We employed perceptual path length (PPL) (Karras et al. 2019) to measure the smoothness of generated samples. Discriminator forest achieves a PPL of 485, which is much better than the PPL of 543 achieved by original StyleGAN2-ADA. And **Figure 4C** shows the generative images and style space interpolation from Forest-GAN and the original model. The generator from Forest-GAN can generate realistic images from intermediate style, however, the one from the original model fails to restore the intermediate mode, which indicates the generator from Forest-GAN captured high generative diversity.

Discussion

In this study, we introduced a discriminative forest GAN (Forest-GAN) framework to improve generative diversity in GANs. The discriminative forest is composed of multiple independent discriminators built on bootstrap sampling data. We have proved that the discriminative forest in Forest-GAN has an upper bound of generalization error that depends on the strength of individual discriminators and the correlation among discriminators. Experimental results demonstrate that a large number of discriminators indeed reduces the generalization error to improve diversity and quality of generated samples. Discriminative forest greatly improves the generative quality and diversity of existing popular GAN models with achieving state-of-the-art performance.

Although a discriminative forest has shown its promising performance in improving generative diversity and quality, several strategies should be explored in future. **(1) De-correlation between individual discriminators.** There is still a need to further reduce the correlation among individual discriminators. De-correlating techniques such as feature bootstrapping, diverse neural network architectures, and adding random noises to input, are worth exploring for their effect on the generalization of discriminative forest in Forest-GAN. **(2) Aggregating function.** In this study, we averaged the gradients of all discriminators to update the generator. GMAN (Durugkar et al. 2017) describes a soft version of one Pythagorean means parameterized λ to aggregate gradients, where $\lambda = 0$ corresponds to *mean* and the *max* is recovered as $\lambda \to \infty$. This kind of aggregating function may work in Forest-GAN for future exploration. **(3) Parallel computing.** Since individual discriminators are independently trained, Forest-GAN can be deployed to a parallel computing paradigm, and thus can be adapted to distributed machine learning.

We also noticed that Nupur Kumari et. al. (Kumari et al. 2021) assembled several pre-trained image classifiers as the discriminator to improve the GAN training. Although they didn't focus on the multi-discriminator adversarial framework, their solution provides a more choice to construct the discriminative forest.

References

Albuquerque, I.; Monteiro, J.; Doan, T.; Considine, B.; Falk, T.; and Mitliagkas, I. 2019. Multi-objective training of Generative Adversarial Networks with multiple discriminators.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Arora, S.; Risteski, A.; and Zhang, Y. 2018. Do GANs learn the distribution? some theory and empirics. In *International Conference on Learning Representations*.

Breiman, L. 2001. Random forests. *Machine learning*, 45(1): 5–32.

Cao, Y.; Li, S.; Liu, Y.; Yan, Z.; Dai, Y.; Yu, P. S.; and Sun, L. 2023. A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT. *ArXiv*, abs/2303.04226.

Chen, J.; Wang, W. H.; Gao, H.; and Shi, X. 2021. PAR-GAN: Improving the Generalization of Generative Adversarial Networks Against Membership Inference Attacks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 127–137.

Choi, J.; and Han, B. 2022. MCL-GAN: Generative Adversarial Networks with Multiple Specialized Discriminators. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Coates, A.; Ng, A.; and Lee, H. 2011. An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In *International Conference on Artificial Intelligence and Statistics*.

Durugkar, I.; Gemp, I.; Mahadevan, S.; et al. 2017. Generative Multi-Adversarial Networks. In *International Conference on Learning Representations*.

Gong, X.; Chang, S.; Jiang, Y.; and Wang, Z. 2019. Autogan: Neural architecture search for generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3224–3234.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Gui, J.; Sun, Z.; Wen, Y.; Tao, D.; and Ye, J. 2021. A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE Transactions on Knowledge and Data Engineering*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Karras, T.; Aittala, M.; Hellsten, J.; Laine, S.; Lehtinen, J.; and Aila, T. 2020. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33: 12104–12114.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2019. Analyzing and Improving the Image Quality of StyleGAN. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8107–8116.

Kumari, N.; Zhang, R.; Shechtman, E.; and Zhu, J.-Y. 2021. Ensembling Off-the-shelf Models for GAN Training. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10641–10652.

Kynkäänniemi, T.; Karras, T.; Laine, S.; Lehtinen, J.; and Aila, T. 2019. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32.

Mordido, G.; Yang, H.; Meinel, C.; et al. 2018. Dropout-GAN: Learning from a Dynamic Ensemble of Discriminators. *ArXiv*, abs/1807.11346.

Nguyen, T.; Le, T.; Vu, H.; and Phung, D. 2017. Dual discriminator generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2670–2680.

Sajjadi, M. S.; Bachem, O.; Lucic, M.; Bousquet, O.; and Gelly, S. 2018. Assessing generative models via precision and recall. *Advances in Neural Information Processing Systems*, 31.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. *Advances in neural information processing systems*, 29.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Thanh-Tung, H.; Tran, T.; and Venkatesh, S. 2019. Improving Generalization and Stability of Generative Adversarial Networks. In *International Conference on Learning Representations*.

Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; and Xiao, J. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

Appendices

A. Global Optimality

We proposed a new discriminative Forest GAN (Forest-GAN) in our main paper. The discriminative forest consists of multiple discriminators which are constructed upon their own bootstrapping datasets. Here we give the global optimality of generator and individual discriminators.

According to the Eq. (5) in main text, for any fixed generator G, the optimal discriminator D_{Θ_k} on its bootstrapping dataset X_k^r with the distribution of $p_{X_k^r}$ is:

$$D^*_{\Theta_k}(x) = \frac{p_{X_k^r}(x)}{p_{X_k^r}(x) + p_g(x)} \ . \tag{A.1}$$

Proof. Given a generator G, the goal of each discriminator D_{Θ_k} is to maximize the quantity $V(G, \{D_{\Theta_k}\}_{k=1}^K)$:

$$\begin{aligned} V(G, \{D_{\Theta_k}\}_{k=1}^K) =& \frac{1}{K}\sum_{k=1}^{K}\{E_{x\sim p_{X_k^r}}[\log D_{\Theta_k}(x)] + E_{z\sim p_z}[\log(1 - D_{\Theta_k}(G(z)))]\} \\ =& \frac{1}{K}\sum_{k=1}^{K}\{E_{x\sim p_{X_k^r}(x)}[\log D_{\Theta_k}(x)] + E_{x\sim p_g}[\log(1 - D_{\Theta_k}(x))]\} \\ =& \frac{1}{K}\sum_{k=1}^{K}\{\int_x p_{X_k^r}(x)\log D_{\Theta_k}(x)dx + \int_x p_g(x)\log[1 - D_{\Theta_k}(x)]dx\} \\ =& \frac{1}{K}\sum_{k=1}^{K}\int_x\{p_{X_k^r}(x)\log D_{\Theta_k}(x) + p_g(x)\log[1 - D_{\Theta_k}(x)]\}dx \ . \end{aligned}$$

According to the proof in (Goodfellow et al. 2014), $\mathcal{F}_k = \int_x\{p_{X_k^r}(x)\log D_{\Theta_k}(x) + p_g(x)\log[1 - D_{\Theta_k}(x)]\}dx$ achieves the maximum when $D_{\Theta_k}(x) = \frac{p_{X_k^r}(x)}{p_{X_k^r}(x)+p_g(x)}$. Since the term $\mathcal{F}_k$ is non-negative and all discriminators $\{D_{\Theta_k}\}_{k=1}^K$ are independent, $V(G, \{D_{\Theta_k}\}_{k=1}^K)$ achieves the maximum if only if all $\{\mathcal{F}_k\}_{k=1}^K$ achieve their maxima. Thus the optimal discriminator $D^*_{\Theta_k}$ is:

$$D^*_{\Theta_k}(x) = \frac{p_{X_k^r}(x)}{p_{X_k^r}(x) + p_g(x)} \ .$$

□

Given K discriminators that have achieved their optimal solutions $\{D^*_{\Theta_k}(x)\}_{k=1}^K$, the globally optimized solution of the generator is achieved if and only if G learns the minimum Jensen-Shannon (JS) divergence of all bootstrapping datasets.

$$\min_G V(G, \{D^*_{\Theta_k}(x)\}_{k=1}^K) \Leftrightarrow \min_{p_g}\sum_{k=1}^{K} JSD(p_{X_k^r} \parallel p_g) \ . \tag{A.2}$$

Proof. The $V(G, \{D^*_{\Theta_k}(x)\}_{k=1}^K)$ can be reformulated as (considering Eq. (A.1)):

$$\begin{aligned} V(G, \{D^*_{\Theta_k}(x)\}_{k=1}^K) =& \frac{1}{K}\sum_{k=1}^{K}\{E_{x\sim p_{X_k^r}}[\log D^*_{\Theta_k}(x)] + E_{z\sim p_z}[\log(1 - D^*_{\Theta_k}(G(z)))]\} \\ =& \frac{1}{K}\sum_{k=1}^{K}\{E_{x\sim p_{X_k^r}}[\log D^*_{\Theta_k}(x)] + E_{x\sim p_g}[\log(1 - D^*_{\Theta_k}(x))]\} \\ =& \frac{1}{K}\sum_{k=1}^{K}[E_{x\sim p_{X_k^r}}\log\frac{p_{X_k^r}(x)}{p_{X_k^r}(x) + p_g(x)} + E_{x\sim p_g}\log\frac{p_g(x)}{p_{X_k^r}(x) + p_g(x)}] \\ =& \frac{1}{K}\sum_{k=1}^{K}[-\log 4 + 2\cdot JSD(p_{X_k^r} \parallel p_g)] \ . \end{aligned}$$

Hence, the globally optimized solution of the generator can be transformed as

$$\min_G V(G, \{D^*_{\Theta_k}(x)\}_{k=1}^K) \Leftrightarrow \min_{p_g}\sum_{k=1}^{K} JSD(p_{X_k^r} \parallel p_g) \ ,$$

where $JSD(\cdot \parallel \cdot) \geq 0$ is the JS divergence between two distributions, and is equal to zero if only if the two distributions are the same. □

B. Upper Bound of the Generalization Error

The generalization error of the discriminative forest D is defined in **Definition 2** in the main paper:

$$\Psi_D = P(\psi(D_\Theta(X,Y)) < 0)\ . \tag{B.1}$$

We derive the upper bound of the generalization error for the discriminative forest based on the margin function defined in the main paper.

Theorem 1. *The upper bound on the generalization error of the discriminative forest D is given by*

$$\Psi_D \leq \bar{\rho}(1-s^2)/s^2\ , \tag{B.2}$$

where $s = E[\psi(D_\Theta(X,Y))]$ is the expectation of margin function, and $\bar{\rho}$ is the mean value of the correlation between two discriminators in the discriminative forest D.

Proof. Assuming $s \geq 0$, Chebychev's inequality [1] gives an upper error bound of Eq. (B.1) :

$$\Psi_D^* \leq \frac{\text{var}[\psi(D_\Theta)]}{s^2} \tag{B.3}$$

According to **Definition 1** in the main paper, the margin function of the discriminative forest is:

$$\begin{aligned}\psi(D_\Theta(X,Y)) =& 2 * P(D_\Theta(X) = Y) - 1\\ =& E_\Theta[2 * I(D_\Theta(X) = Y) - 1]\ .\end{aligned} \tag{B.4}$$

Let $\psi_{raw}(D_\Theta(X,Y)) = 2I(D_\Theta(X) = Y) - 1$, denoting the raw margin function of individual discriminators. The margin function $\psi(D_\Theta(X,Y))$ is the expectation of $\psi_{raw}(D_\Theta(X,Y))$. Thus,

$$\begin{aligned}[\psi(D_\Theta(X,Y))]^2 =& [E_\Theta(\psi_{raw}(D_\Theta(X,Y)))]^2\\ =& E_{\Theta,\Theta'}\{\psi_{raw}[D_\Theta(X,Y)]\psi_{raw}[D_{\Theta'}(X,Y)]\}\ ,\end{aligned} \tag{B.5}$$

where Θ, Θ' are independent and identity distributions [2].

Then, we have

$$\begin{aligned}\text{var}[\psi(D_\Theta)] =& E_{\Theta,\Theta'}\{\text{cov}[\psi_{raw}(D_\Theta)\psi_{raw}(D_{\Theta'})]\}\\ =& E_{\Theta,\Theta'}[\rho_{\Theta,\Theta'}\text{std}(\psi_{raw}(D_\Theta))\text{std}(\psi_{raw}(D_{\Theta'})]\ ,\end{aligned} \tag{B.6}$$

where $\rho_{\Theta,\Theta'}$ is the correlation between $\psi_{raw}(D_\Theta)$ and $\psi_{raw}(D_{\Theta'})$, and $\text{std}(\psi_{raw}(D_\Theta))$ is the standard deviation of $\psi_{raw}[D_\Theta(X,Y)]$ with fixed Θ and Θ'.

Let $\bar{\rho} = E_{\Theta,\Theta'}[\rho_{\Theta,\Theta'}\text{std}(\psi_{raw}(D_\Theta))\text{std}(\psi_{raw}(D_{\Theta'})]/E_\Theta[\text{std}(\psi_{raw}(D_\Theta))\text{std}(\psi_{raw}(D_{\Theta'})]$, denoting the mean correlation of D_Θ and $D_{\Theta'}$. Then

$$\begin{aligned}\text{var}[\psi(D_\Theta)] =& \bar{\rho}E_\Theta[\text{std}(\psi_{raw}(D_\Theta))\text{std}(\psi_{raw}(D_{\Theta'})]\\ =& \bar{\rho}E_\Theta(\text{var}(\psi_{raw}(D_\Theta)))\ .\end{aligned} \tag{B.7}$$

$E_\Theta(\text{var}(\psi_{raw}(D_\Theta)))$ can be rewritten as[3]:

$$\begin{aligned}E_\Theta(\text{var}(\psi_{raw}(D_\Theta))) =& E_\Theta\{E(\psi_{raw}(D_\Theta))^2 - [E(\psi_{raw}(D_\Theta))]^2\}\\ =& E_\Theta[E(\psi_{raw}(D_\Theta))^2] - s^2\\ \leq& 1 - s^2\ .\end{aligned} \tag{B.8}$$

Combining Eq. (B.3), (B.7) and (B.8) together yields the upper bound of generalization error:

$$\Psi_D^* \leq \frac{\bar{\rho}(1-s^2)}{s^2}\ . \tag{B.9}$$

□

C. Implementation Details

Model Architecture

We evaluated Forest-GAN on one simulation dataset and two popular image datasets (i.e. STL10 (Coates, Ng, and Lee 2011) and LSUN-Cat (Yu et al. 2015)). Model architecture and hyper-parameters for simulation dataset are listed in **Table C1**, and this experiment is implemented with Tensorflow 2.0. The basis model architecture and hyper-parameters for STL10 (Coates, Ng, and Lee 2011) is AutoGAN (Gong et al. 2019), and the basis model architecture and hyper-parameters for LSUN-Cat (Yu et al. 2015)) is StypleGAN2-ADA(Karras et al. 2020). We only increased the number of discriminators according to the discriminative forest framework settings. All experiments are trained on a computing server equipped with Intel(R) Xeon(R) Gold 6248R CPU @ 3.00GHz and NVIDIA A100 GPU.

[1] Chebychev's inequality: $P\{|X-\mu| \geq \epsilon\} \leq \sigma^2/\epsilon^2$.

[2] For any function f and parameters Θ, Θ' with an independent distribution, $[E_\Theta(f_\Theta)]^2 = E_{\Theta,\Theta'}(f_\Theta f_{\Theta'})$.

[3] For any distribution, the variation $\text{var}(x) = E(x^2) - [E(x)]^2$.

Table C1: Model architecture of Forest-GAN on simulation data. An Adam optimizer with $\beta_1 = 0.5, \beta_2 = 0.99$ is used. Learning rate is set at 10^{-5}.

Generator	Discriminator
InputNoise(20) Dense(units=256) LeakReLU() Dense(units=256) LeakyReLU() Dense(units=2)	InputPoints(2) Dense(units=256) LeakyReLU() Dense(units=256) LeakyReLU() Dense(units=2) Sigmoid()

Modified Precision and Recall

Since the Precision and Recall metrics in (Kynkäänniemi et al. 2019) don't consider distribution density in the feature space, we modified the conditional function by adding a density constraint.

The real and generated samples are drawn from $X^r \sim p_r$ and $X^g \sim p_g$, respectively. To accurately estimate the similarity between samples, the real and generated samples are mapped into a feature space. The feature vectors of the real and generated images are denoted as ϕ_r and ϕ_g, and the corresponding sets of feature vectors are Φ_r and Φ_g. An equal number of samples are sampled from each distribution, in other words, $|\Phi_r| = |\Phi_g|$. Given a sample ϕ, its binary function is modified as:

$$f(\phi, \Phi) = \begin{cases} 1, \text{ if } ||\phi - \phi'||_2 \leq ||\phi' - NN_k(\phi', \Phi)||_2 \text{ for at least one } \phi' \in \Phi \text{ and } \lambda \geq \frac{d_g}{d_r} \geq \frac{1}{\lambda}, \\ 0, \text{ otherwise }, \end{cases} \tag{C.1}$$

where $NN_k(\phi', \Phi)$ returns the i-th nearest feature vector of ϕ' from set Φ. d_g and d_r are the radius of the hypersphere in real and generated space respectively. λ is a hyper-parameter which limits the difference of d_g and d_r. The Precision and Recall values are calculated in the same way as in (Kynkäänniemi et al. 2019). Precision is quantified by querying for each generated sample to evaluate whether the sample is within the estimated manifold of real feature space. Symmetrically, Recall is calculated by querying each real sample to assess whether the sample is within the estimated manifold of generated feature space.

$$precision(\Phi_r, \Phi_g) = \frac{1}{|\Phi_g|} \sum_{\phi_g \in \Phi_g} f(\phi_g, \Phi_r), recall(\Phi_r, \Phi_g) = \frac{1}{|\Phi_r|} \sum_{\phi_r \in \Phi_r} f(\phi_r, \Phi_g) \tag{C.2}$$

In this study, for the simulation data, as it has only two dimensions (2D), we directly identified neighbors in a 2D feature space. For image datasets (STL10 (Coates, Ng, and Lee 2011) and LSUN-Cat (Yu et al. 2015)), we embedded the images into a high-dimensional feature space using a pre-trained Inception-V3 network (Simonyan and Zisserman 2014). $\lambda = 3$ was used in all experiments.

D. Auxiliary Results

Improving training stability and model convergence on simulated data

Here, we provided extended results to illustrate the advantages of a larger K on training stability and model convergence in Forest-GAN. **Figure D1** shows the training loss curves of Forest-GAN with a vanilla GAN loss on simulation data. When K is small, the training loss curves are undulating and thus unstable. As K increases, the training stability of the generator and all individual discriminators are improved. We also observe that Forest-GAN with $K = 50$ converges within 1,000 epochs, while Forest-GAN with $K = 1$ doesn't converge during the whole training process. These results imply that larger K improves the training stability and accelerates the convergence of Forest-GAN.

Parameter settings and performance comparison of AutoGAN on STL10

As mentioned in the main text, we scaled the width of each layer in AutoGAN to control the total parameters in discriminator forest. Table D1 shows total trainable parameters in each discriminator and whole discriminative forest.

Figure D2 shows the performance of AutoGAN improved by discriminative forest on STL10 in terms of four metrics: FID, IS, Precision and Recall.

The generated images comparison of StyleGAN2-ADA on LSUN_Cat

We used the generators from the original StyleGAN2-ADA and Forest-GAN to randomly generate 100 images, respectively, and select 10 images from each of them for display. Figure D3 shows the results of Forest-GAN with K=5, and Figure D4 is results of the original StyleGAN2-ADA. The images generated by our Forest-GAN are more realistic and can pay more attention to the details of the subject, including contour, beards and etc.

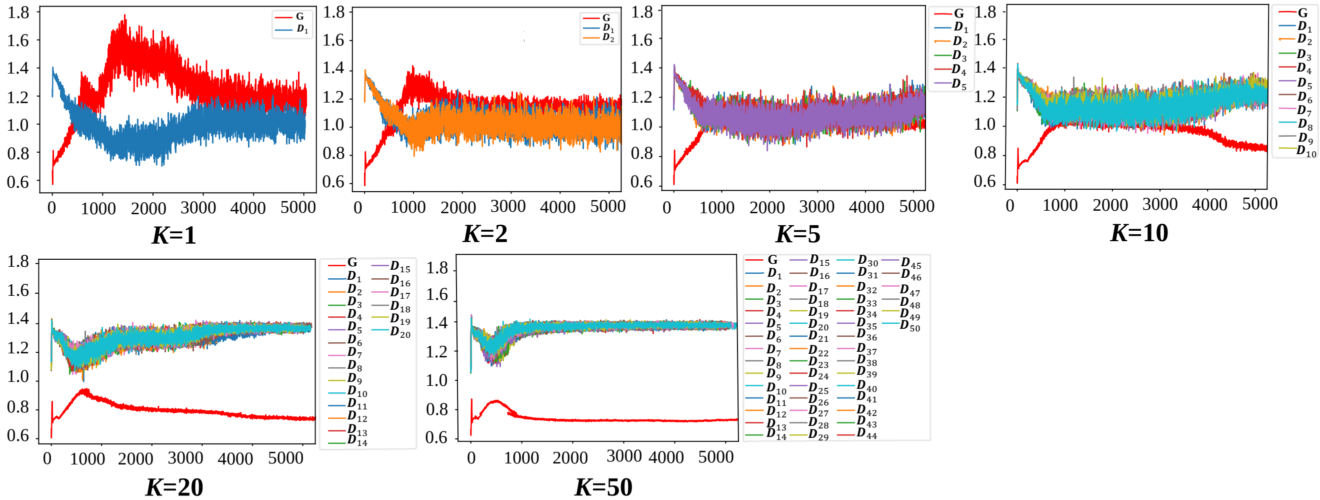


Figure D1: Training loss of Forest-GAN on simulation data. Larger K improves the training stability of both generator and individual discriminators, and model convergence is faster when K is larger.

Table D1: The trainable parameters in each discriminator and whole discriminative forest. The total amount of parameters in the discriminator forest is controlled through two hyperpatameters, K and df_dim. '*' indicates the parameter setting in original discriminator of AutoGAN.

Para (in each discriminator) (in whole discriminator forest)	K Trainable Parameters	df_dim	Trainable Parameters	
1M	* 1	128	1,053,824	1,053,824
	2	92	545,468	1,090,936
	5	58	217,674	1,088,370
	10	42	114,618	1,146,180
2M	1	186	2,107,648	2,221,770
	2	128	1,053,824	2,107,648
	5	80	412,880	2,064,400
	10	58	217,674	2,176,740
5M	1	288	5,320,224	5,320,224
	2	200	2,568,200	5,136,400
	5	128	1,053,824	5,269,120
	10	90	522,090	5,220,900
10M	1	400	10,256,400	10,256,400
	2	288	5,320,224	10,640,448
	5	178	2,035,074	10,175,370
	10	128	1,053,824	10,538,240

To empirically evaluate the generative diversity of our generator, we made style space interpolation for the generators of original and our Forest-GAN, respectively, as shown in Figure D6 and Figure D5. We performed multi-style interpolation on the generted images, including color transitions, action transitions and demeanor transitions. We first generated images at both ends, and then calculated the intermediate latent noise between them to synthesize the intermediate image.

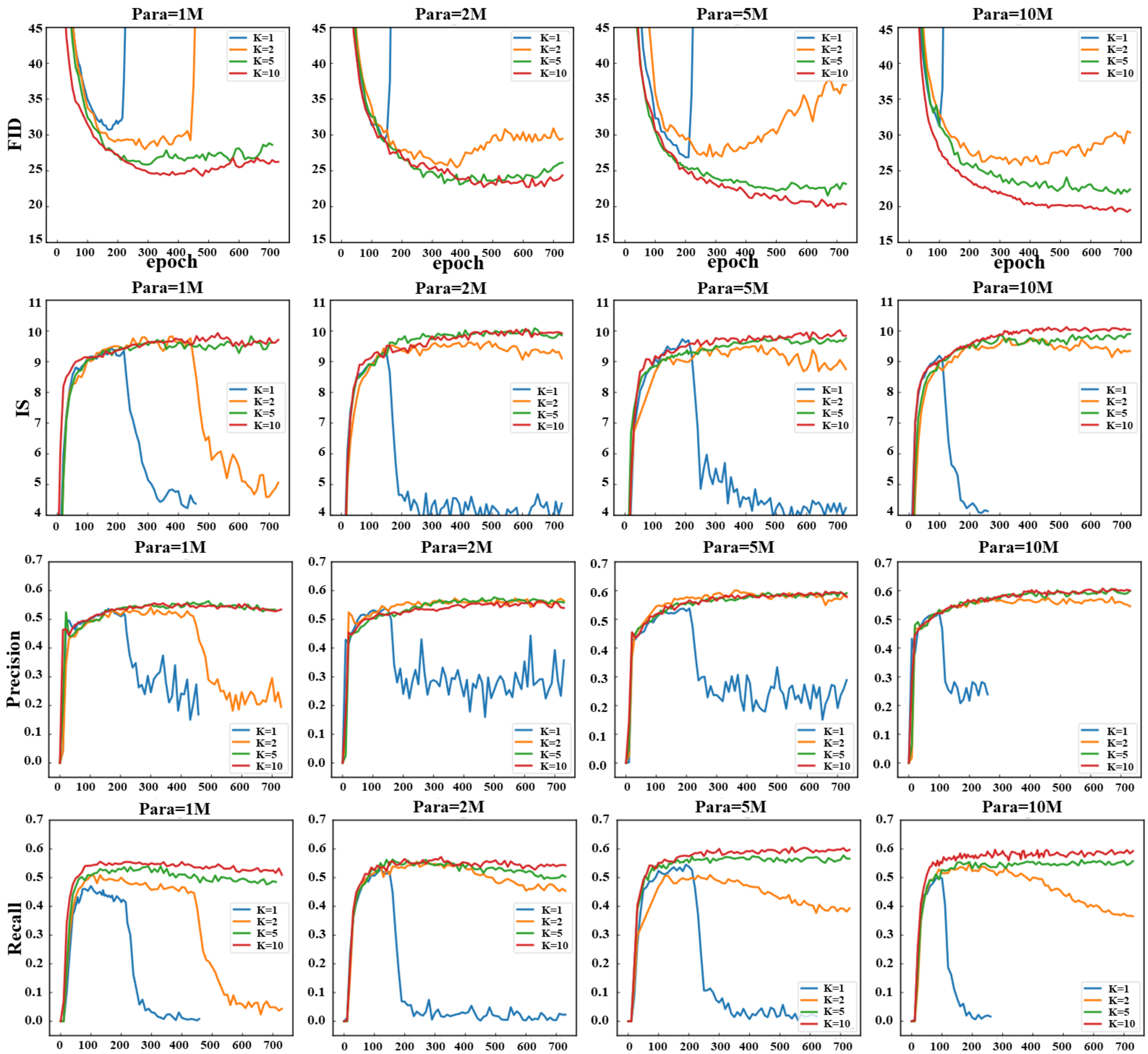


Figure D2: The performance of AutoGAN improved by discriminative forest on STL10. Discriminative forest can stabilize the training process in all parameter settings as K increased in discriminator forest

Figure D3: The images generated by Forest-GAN with K=5.

Figure D4: The images generated by original StyleGAN2_ADA.

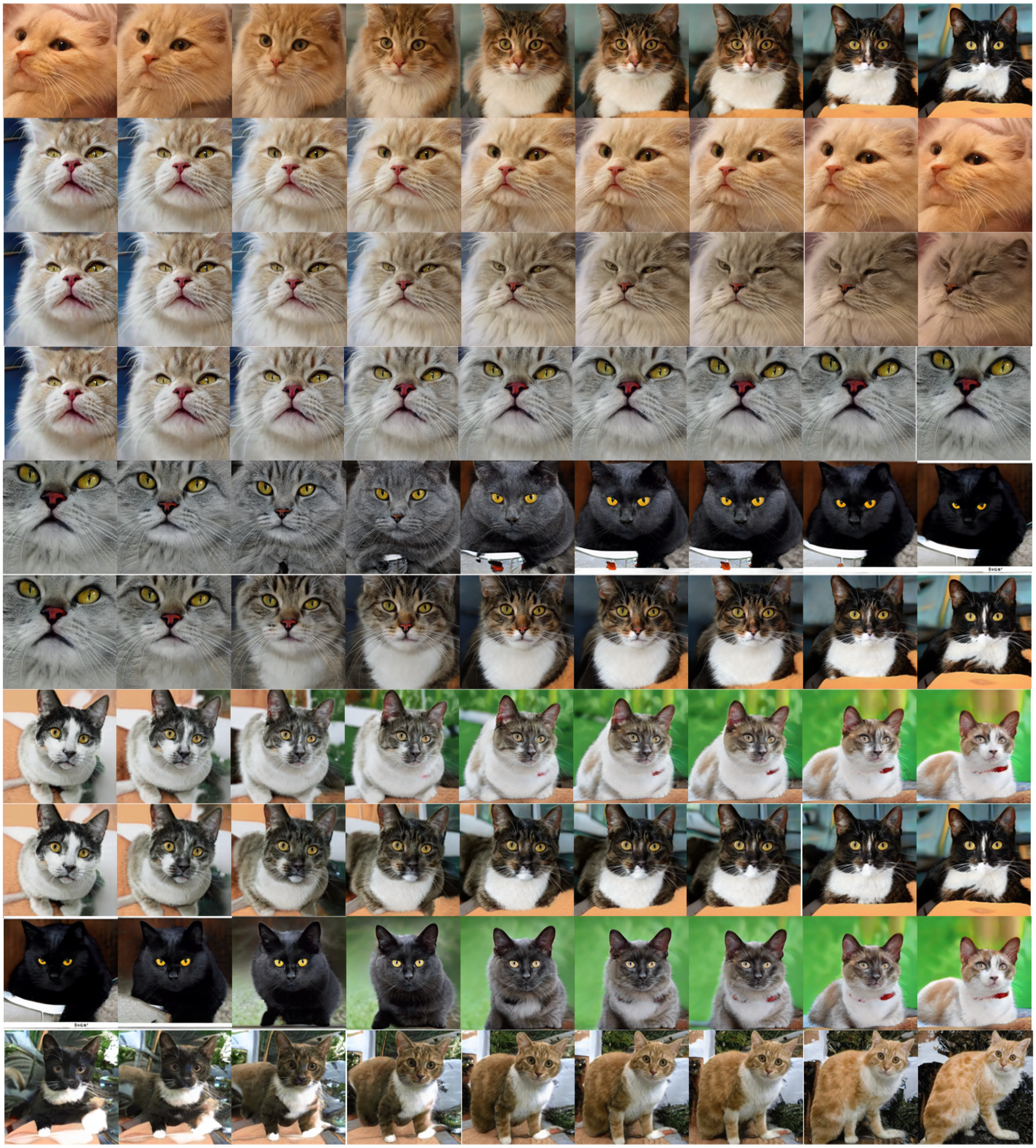


Figure D5: The style space interpolation of Forest-GAN with K=5. Forest-GAN enables smooth interpolation between style latent embeddings of two generated images.

Figure D6: The style space interpolation of original StyleGAN2_ADA. Although it can realize the interpolation operation, some intermediate modes cannot be correctly mapped into valid images.