FreGAN: Exploiting Frequency Components for Training GANs under Limited Data

Mengping Yang[†‡] **Zhe Wang**[†‡*] **Ziqiu Chi**[†‡] **Yanbing Zhang**[†‡]
[†]Department of Computer Science and Engineering, ECUST, China
[‡]Key Laboratory of Smart Manufacturing in Energy Chemical Process, ECUST, China
wangzhe@ecust.edu.cn
mengpingyang@mail.ecust.edu.cn

Abstract

Training GANs under limited data often leads to discriminator overfitting and memorization issues, causing divergent training. Existing approaches mitigate the overfitting by employing data augmentations, model regularization, or attention mechanisms. However, they ignore the frequency bias of GANs and take poor consideration towards frequency information, especially high-frequency signals that contain rich details. To fully utilize the frequency information of limited data, this paper proposes FreGAN, which raises the model's frequency awareness and draws more attention to producing high-frequency signals, facilitating high-quality generation. In addition to exploiting both real and generated images' frequency information, we also involve the frequency signals of real images as a self-supervised constraint, which alleviates the GAN disequilibrium and encourages the generator to synthesize adequate rather than arbitrary frequency signals. Extensive results demonstrate the superiority and effectiveness of our FreGAN in ameliorating generation quality in the low-data regime (especially when training data is less than 100). Besides, FreGAN can be seamlessly applied to existing regularization and attention mechanism models to further boost the performance. [2]

1 Introduction

Generative adversarial networks (GANs) [11] have shown impressive achievements in synthesising plausible and photorealistic visual objects, such as image [19] [18] and video [42] generation, image inpainting [26], image translation [38] and so on. However, a prerequisite of such success is sufficient training data, which impedes applications of GANs in areas where only dozens of data are available or where it is challenging to collect massive data due to geographical, spatial, temporal, or privacy reasons. Thus developing data-efficient GANs that can generate plausible images under limited data, without compromising the quality, is necessary and meaningful.

Training GANs under limited data often leads to overfitting and instability issues [17] [15]. Specifically, when the discriminator (D) overfits to the limited training data, it simply remembers the input real images and classifies others as fake images, thus providing meaningless feedback to the generator (G), leading to divergent training and poor-quality generation. Ameliorating the synthesize quality under limited data is still an unexplored problem. Recent approaches for this problem include enlarging the training set with different data augmentations [55] [39] [54] [17] [15], regularizing the output of D with an additional constraint [40], and devising new network architectures [25]. However, existing methods are mainly developed from the perspective of data scale and model capacity, and they ignore

[*]Corresponding author
[2]Our codes are available at https://github.com/kobeshegu/FreGAN_NeurIPS2022.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

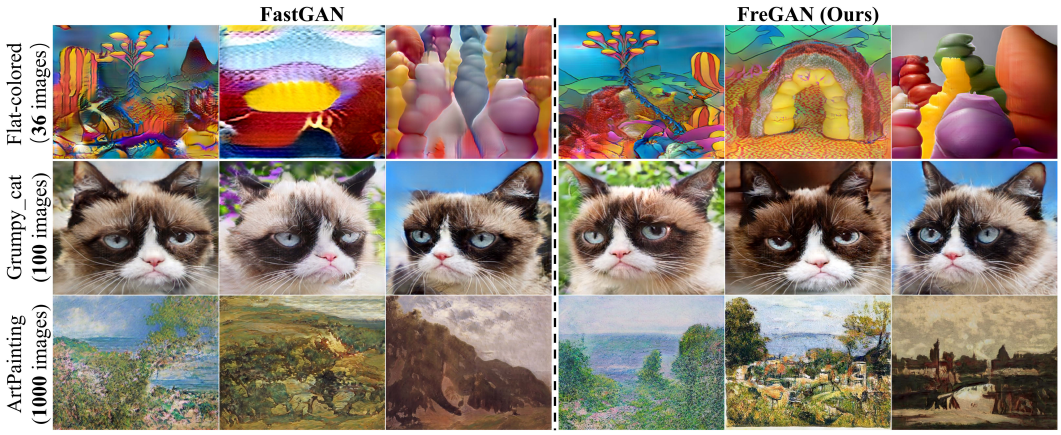


Figure 1: **Images generated by FastGAN [25] and our FreGAN given limited training data.** The details of FastGAN deteriorate while our FreGAN effectively ameliorates the synthesize quality by raising the model's frequency awareness, producing plausible images with better fine details.

a critical property of the data itself, *i.e.*, frequency signals. GANs have been demonstrated to have a spectral bias in fitting frequency signals [31] [37]. They preferentially fit low-frequency signals and tend to ignore high-frequency signals [48], which encode fine details like vertical and horizontal edges [50] [10]. Missing them may lead to unrealistic image synthesize with unsatisfactory artifacts (see Fig. 1). This paper proposes a frequency-aware model, termed as FreGAN, to raise the frequency awareness of G and D. By encouraging G to generate more reasonable and adequate high-frequency signals, our FreGAN ameliorates the synthesize quality under limited data, as shown in Fig. 1.

To fully exploit the frequency information of limited training data, we first decompose images into different frequency components via Haar wavelet transformation [6]. Unlike traditional wavelet transformation that is employed at the image level, we perform it on the intermediate features of both D and G. We then employ a high-frequency discriminator (HFD) and frequency skip connection (FSC) to raise the frequency awareness of G and D, respectively. However, G still has no explicit clue about what high-frequency signals it should synthesize, and D is overconfident in making real/fake decisions after seeing real and fake images. Such an unbalanced competition motivates us to perform high-frequency alignment (HFA) to alleviate the information asymmetry between G and D. Innovatively, we explicitly exploit the frequency signals of real images induced from D as a self-supervised constraint to guide G to leverage the frequency knowledge properly. Besides, HFD and HFA are applied on multi-scale features to thoroughly excavate the frequency signals of limited data, mitigating frequency bias and loss of high-frequency information.

The primary contributions of this paper are three-fold: 1) we propose FreGAN to raise the model's frequency awareness, which successfully mines and exploits frequency information of limited data, and as a byproduct, FreGAN alleviates the unhealthy competition between G and D; 2) we demonstrate the compatibility of our model by combining our method with other techniques like regularization [40] and attention mechanism [23]; 3) we perform extensive experiments on various datasets with limited data, and our FreGAN achieves state-of-the-art performance on these datasets, indicating the effectiveness and superiority of our method for ameliorating synthesize quality, especially when training data is extremely limited.

2 Related Work

Generative Adversarial Networks. Generative adversarial networks (GANs) [47] [16] [12], which target at generating plausible and realistic images, have made massive progress since the pioneering work [11]. The capability of GANs enables various visual applications like image [19] [20] and video generation [42] [41], image inpainting [51] [26], image manupulation [10] [38] and super-resolution [45], etc. However, GANs are notoriously difficult to train as several issues like mode collapse and instability happen easily. Numerous techniques have been proposed to stabilize training and improve the synthesize quality by designing new optimization objectives or network architectures [14]. WGAN [2] and f-GAN [29] minimize the Wasserstein distance and the f-divergence of real and generated distribution instead of minimizing JS divergence in [11]. BigGAN [7] and

StyleGAN series [16] [19] [20] [17] [18] have made breakthrough progress in producing realistic images. However, the performance of these models deteriorates when given limited data.

Training GANs under limited data. Improving the synthesize quality under limited data remains an underexplored problem, which has drawn extensive attention recently. Insufficient training data leads to discriminator overfitting, thus degrading the quality of generated images. One straightforward way to address such data scarcity is to expand the training set with various augmentations. In addition to employing conventional augmentation techniques [39] [55] (e.g., flip, crop), ADA [20] and DiffAug [54] propose adaptive and differentiable augmentation to enlarge the training data, respectively. APA [15] deceives D based on the degree of overfitting with an adaptive pseudo augmentation. InsGen [49] involves instance discrimination as an auxiliary task to encourage D to distinguish every individual image, which improves the discriminative power of the discriminator. Lecam [40] regularizes the output of the discriminator throughout the training process. FastGAN [25] employs a skip-layer channel-wise excitation module and a self-supervised discriminator to stabilize and accelerate the training. The most recently MoCA [23] improves few-shot image generation quality with a prototype memory with an attention mechanism. Benefit from the significant progress of large-scale pre-trained visual recognition models, Vision-aided GAN [21] uses available off-the-shelf models to help the GAN training and ProjectedGAN [33] improve GANs by projecting generated and real images into pre-trained feature spaces. Another category of methods transfer and reuse knowledge from models that are pre-trained on large-scale data, *i.e.*, few-shot GAN adaptation [44] [24] [43] [30]. In this paper, we ameliorate the synthesize quality under limited data from the frequency domain perspective. By raising the frequency awareness of GANs and providing more fine details to G, we facilitate photorealistic image generation. Our work is complementary to previous model regularization and attention mechanism approaches, and our method promotes equilibrium between G and D.

Wavelet Transformation in GANs. Schwarz et al. [34] prove that GANs exhibit a frequency bias and resolving frequency artifacts is necessary for photorealistic image generation. Consequently, GANs tend to ignore high-frequency signals as they are hard to generate, compromising the generation quality. Wavelet transformation [6], which decomposes images into frequency components with different bands, has been wildly used in various applications of GANs, such as style transfer [4] [50], image inpainting [51], image editing [10], etc. HiFA [10] alleviates the generator's pressure of producing high-frequency signals by directly feeding high-frequency components to the generator. WaveFill [51] disentangles different frequency signals and explicitly fills the missing regions in each frequency band, achieving superior image inpainting. Zhang et al. [52] propose wavelet knowledge distillation towards efficient image-to-image translation without a performance drop. SWAGAN [9] incorporates wavelet with the hierarchical training of StyleGAN2 [20] and performs wavelets at the image level. Our FreGAN is more flexible by directly decomposing intermediate features of the generator and the discriminator into the wavelet domain, and no additional down/up sampling are required to convert images to higher/lower resolution as in SWAGAN, which makes our method more efficient. Unlike existing methods that are performed on ample data, this paper addresses the more challenging few-shot generation problem. In addition to raising the frequency awareness of the model, we also mitigate the unhealthy competition by lessening the frequency gap between G and D.

3 Methodology

The overall framework of our FreGAN is illustrated in Fig. 2. To formulate our method, we explicitly utilize wavelet transformation to decompose features into different frequency components. We then employ high-frequency discriminator (HFD) and frequency skip connection (FSC) to raise the frequency awareness of G and D, respectively. Moreover, we perform high-frequency alignment (HFA) to further guide G to synthesize adequate frequency signals.

3.1 Wavelet Transformation

To decompose images into different frequency components, we adopt a simple but effective wavelet transformation, *i.e.*, Haar wavelet. Haar wavelet consists of two mirror operations: wavelet pooling and wavelet unpooling. The former converts images into the hlwavelet domain, and the latter inversely reconstructs frequency components into the spatial domain. There are four kernels in wavelet pooling

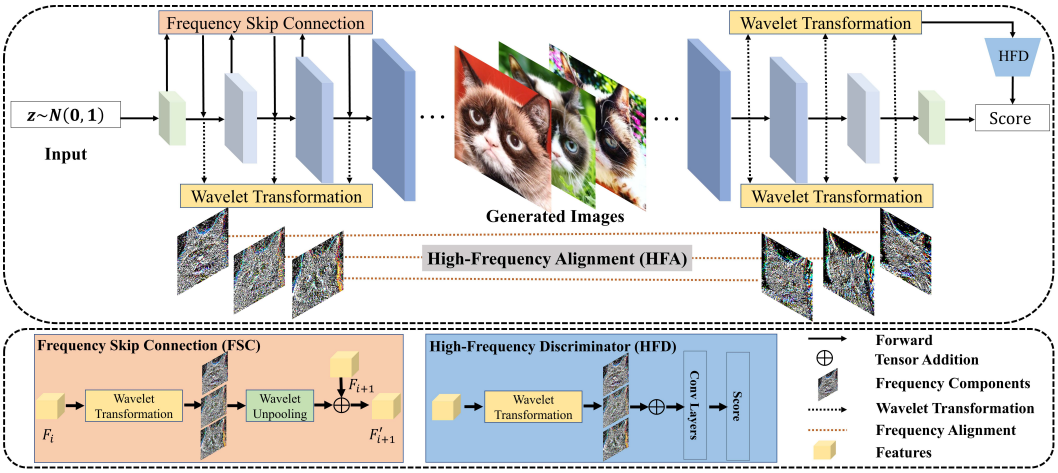


Figure 2: **The overall framework of our proposed FreGAN.** Composed of three key ingredients, *i.e.*, frequency skip connection (FSC), high-frequency discriminator (HFD), and high-frequency alignment (HFA), our FreGAN raises the model's frequency awareness, facilitating high-quality image synthesize under limited data.

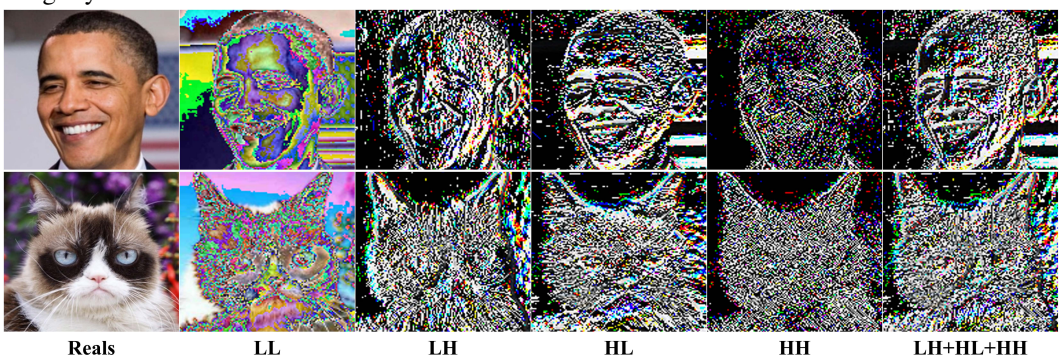


Figure 3: **Illustration of frequency components obtained from Haar wavelet transformation.** The low (L) pass filter captures images' overall textures and outlines, and the high (H) pass filter concentrates on details such as the background and edges.

operation: LL^T, LH^T, HL^T, HH^T, where $L^T = \frac{1}{\sqrt{2}}[1,1]$, $H^T = \frac{1}{\sqrt{2}}[-1,1]$, L and H denotes the low and high pass filters, respectively. The low (L) pass filter captures the outline and surface of images, while the high (H) pass filter focuses on detailed information like the edges and delicate textures. Fig. 3 illustrates the obtained frequency components of given images via Haar wavelet. We can observe that low-frequency component LL contains the overall surface of images, while components that are decomposed by high pass filters, i.e., LH, HL, HH, contain more fine details. Further, by summing up the three high-frequency components, we approximately obtain all details information of images, *e.g.*, the eyes of the cat and the teeth of Obama.

3.2 High-Frequency Discriminator

To raise the frequency awareness of D, we devise high-frequency discriminator (HFD). HFD is responsible for distinguishing the real images from the generated images from the perspective of the frequency domain. Formally, for the i-th layer in the discriminator, we adopt wavelet pooling on the intermediate features and obtain $LL_D^i, LH_D^i, HL_D^i, HH_D^i$, then we combine the three high-frequency components by tensor addition, *i.e.*, $HF_D^i = LH_D^i + HL_D^i + HH_D^i$, which contains sufficient details of features. By applying traditional convolution and downsampling operations following the original discriminator, we define the adversarial loss of our HFD as:

$$\mathcal{L}_D^{HF} = -\mathbb{E}_{HF_{\text{real}} \sim I_{\text{real}}}[\min(0, -1 + D_H(HF_{\text{real}}))] - \mathbb{E}_{HF_{\text{fake}} \sim G(z)}[\min(0, -1 - D_H(HF_{\text{fake}})] \quad (1)$$

$$\mathcal{L}_G^{HF} = \mathbb{E}_{HF_{\text{fake}}}[D_H(HF_{\text{fake}})] \quad (2)$$

where HF_{real} and HF_{fake} are the high-frequency information of real and fake images, respectively. D_H is the high-frequency discriminator. Since the high-frequency information may be eschewed by D as the network goes deeper, we perform multi-scale HFD on multi-layers of the discriminator. The multi-scale operation ensures fully mine and exploit the frequency information of limited data, which further improves D's frequency awareness. Notably, being guided by the HFD with Eq. 2, G is also optimized to produce rich high-frequency details.

3.3 Frequency Skip Connection

The generator is capable of producing plausible frequency signals after employing HFD (see Tab. 5). However, as GANs fit frequency signals from low to high and the high-frequency signals may be ignored as the network goes deeper. To prevent the loss of high-frequency information and further encourage the generator to produce rich details, we propose frequency skip connection (FSC). Concretely, we utilize wavelet unpooling operation of the frequency components $LL_G^i, LH_G^i, HL_G^i, HH_G^i$ obtained from wavelet transformation on the features of G's i-th layer, which reconstructs the high-frequency representation to the original features. Then we explicitly feed the reconstructed frequency representations to the next layer of G. Formally,

$$F_{i+1}^{'} = F_{i+1} + Unpooling(LL_G^i, LH_G^i, HL_G^i, HH_G^i) \tag{3}$$

where F_i denotes the features of the i-th layer and $Unpooling$ is the wavelet unpooling operation. $F_{i+1}^{'}$ is the obtained features after FSC, which will be fed into the subsequent layer for further operation. Such skip connection prevents loss of high-frequency information and maintains high-frequency details.

3.4 High-Frequency Alignment

Adding HFD and FSC explicitly raises the frequency awareness of G, but G can only synthesize arbitrary frequency signals. How G can utilize the frequency signals is still ambiguous, and D still dominates the competition since it learns discriminative knowledge from both real and generated images. To balance the unhealthy competition between G and D, we propose high-frequency alignment (HFA), which involves high-frequency signals of real images induced from D as a regularizer to guide G, promoting G to synthesize more reasonable and realistic fine details. Specifically, we extract the frequency representations of intermediate features of G at different layers. For the i-th layer of G, we obtain frequency components $LL_G^i, LH_G^i, HL_G^i, HH_G^i$. We ignore LL_G^i and combine the three high-frequency components, *i.e.*, $HF_G^i = LH_G^i + HL_G^i + HH_F^i$. Then we use the high-frequency components of the discriminator HF_D^i as a self-supervision constraint. In addition to fool D, G is expected to minimize the distance of high-frequency information between the generated and real images. The alignment loss is defined as:

$$\mathcal{L}_{\text{align}} = \|HF_D - HF_G\|_1 \tag{4}$$

where $\| * \|_1$ denotes the L_1-norm. Such alignment encourages G to synthesis frequency signals that approach real frequency signals, mitigating the unhealthy competition and facilitating generation quality. To take full advantage of frequency signals of real images from D, we perform HFA on multi-scale features like HFD as shown in Fig. 2. The ablative experiment results in Sec. 4.2 demonstrate the rationality and effectiveness of employing HFA and HFD on multi-scale features.

3.5 Optimization

Following [25], we adopt the hinge version of adversarial loss to train our model.

$$\mathcal{L}_D = -\mathbb{E}_{x\sim I_{\text{real}}}[\min(0, -1 + D(x))] - \mathbb{E}_{\hat{x}\sim G(z)}[\min(0, -1 - D(\hat{x})] \tag{5}$$

$$\mathcal{L}_G = -\mathbb{E}_{z\sim\mathcal{N}}[D(G(z))] \tag{6}$$

We also use the reconstruction loss [25] to encourage the discriminator to extract more representative features.

$$\mathcal{L}_{\text{recons}} = \mathbb{E}_{\mathbf{f}\sim D_{\text{encode}}(x), x\sim I_{\text{real}}}[\|\mathcal{G}(\mathbf{f}) - \mathcal{T}(x)\|] \tag{7}$$

where $\mathbf{f}$ is the intermediate features of D, $\mathcal{G}$ and $\mathcal{T}$ denote the processing on the features $\mathbf{f}$ and the input images x. In sum, our discriminator is optimized by $\mathcal{L}_D$, $\mathcal{L}_{\text{recons}}$, and $\mathcal{L}_D^{HF}$. Our generator is optimized by $\mathcal{L}_G$, $\mathcal{L}_G^{HF}$ and $\mathcal{L}_{\text{align}}$, the coefficient of each loss is set to 1.

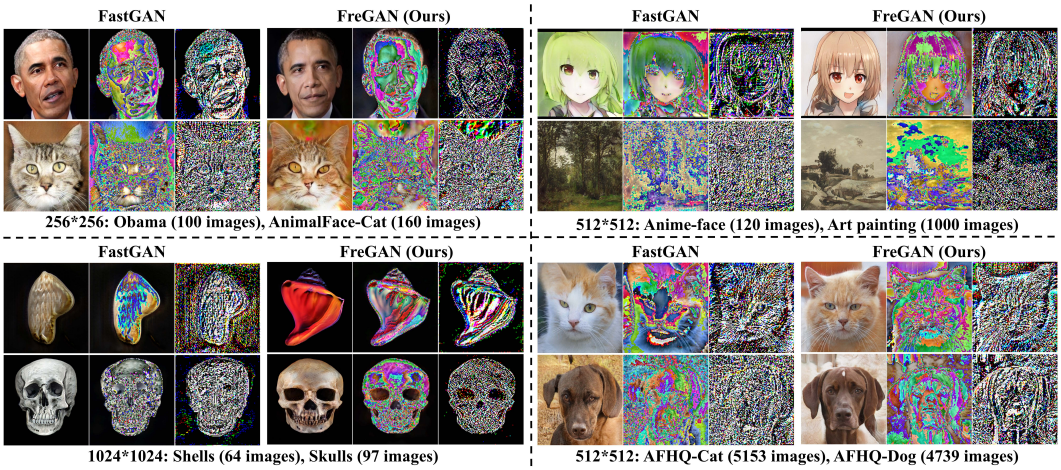


Figure 4: **Qualitative comparison results of our FreGAN and baseline FastGAN.** The images from left to right are generated images, low-frequency, and high-frequency components, respectively. Our FreGAN improves the overall quality of generated images and raises the model's frequency awareness, encouraging the generator to produce precise high-frequency signals with fine details.

4 Experiments

Datasets. We test the effectiveness of our method on low-shot datasets from various domains with different resolutions. On 256 $\times$ 256 resolution, we use Animal Face Dogs and Cat [35], as well as 100-shot-Panda, Obama, and Grumpy_cat [54]. On 512 $\times$ 512 resolution, we use Anime-Face, Art Paintings, Moongate, Flat-colored, and Fauvism-still-life [25]. On 1024 $\times$ 1024 resolution, we use Pokemon, Skulls, Shells, MetFace [17] and BrecaHAD [1]. These datasets contain a limited number of samples (mostly less than 1,000) and cover art paintings, realistic photos, human faces, etc. For datasets that are not strictly equal to the corresponding resolution, we resize them to the closest resolution in implementation. Besides, we use AnimalFace HQ (AFHQ) datasets [5] to evaluate the performance of our model when training with more data ($\sim$5k).

Evaluation metrics and baseline. We adopt two common metrics to evaluate the synthesize quality: Fréchet Inception Distance (FID) [13] and Kernel Inception Distance (KID) [3]. The lower FID and KID is, the better the generation quality is. FID quantifies the distance between the distribution of the generated and the real images. KID, which is designed unbiased, has been proven more descriptive for small datasets [17], note that all KID scores reported in our paper need to $\times 10^{-3}$ following [17]. Following [25], we calculate FID and KID by measuring the distance between all available training images and 5k generated images. We also provide the LPIPS [53], IS [32], Precision, Recall [22], Density, Coverage [28] results in the appendix.

We compare our model with: 1) the state-of-the-art generative model StyleGAN2 [20], and SWAGAN [9], which incorporates wavelet into StyleGAN2; 2) data augmentation-based approaches that is designed for training GANs with limited data, *i.e.*, ADA [17], DiffAug [54], APA [15]; 3) the state-of-the-art few-shot generative model FastGAN [25]. We reimplement all baselines with their released official code under consistent settings for a fair comparison. Implementation details of baseline models are given in the appendix.

Implementation details. We choose the current state-of-the-art few-shot generative model FastGAN [25] as the backbone and implement our proposed techniques upon it. all other settings remain the same as [25]. We decompose the intermediate 8 $\times$ 8, 16 $\times$ 16, 32 $\times$ 32 features of G and D into frequency components for our frequency skip connection, high-frequency discriminator, and high-frequency alignment. More implement details are given in the appendix.

4.1 Main Results

Quantitative comparison on datasets with limited data amounts. The quantitative comparison results of our FreGAN and baseline methods on different resolutions are given in Tab. 1, Tab. 2 and Tab. 3. We save the best training snapshots of each method and generate 5k images to compute FID and KID. The whole training set is adopted as the referenced distribution. We can observe from the

Table 1: The FID (lower is better) and KID (lower is better) scores of our method compared to state-of-the-art methods on $\mathbf{256 \times 256}$ **datasets with limited data amounts**.

	Animal Face				100-shot					
	Dog (389 imgs)		Cat (160 imgs)		Panda		Obama		Grumpy_cat	
Method	FID	KID	FID	KID	FID	KID	FID	KID	FID	KID
StyleGAN2 [20]	113.86	91.31	79.04	34.43	18.05	7.40	69.01	52.63	35.00	11.01
SWAGAN [9]	82.47	80.46	59.71	19.35	27.55	11.92	71.05	55.82	38.44	17.03
ADA [17]	55.48	18.42	37.95	6.43	14.17	6.53	43.17	13.23	43.80	45.01
APA [15]	81.16	26.42	42.60	7.97	19.21	10.80	42.97	15.71	28.10	5.53
DiffAug [54]	61.34	24.51	41.84	12.27	11.52	3.57	48.85	23.31	26.89	9.42
FastGAN [25]	52.46	18.22	33.85	4.99	9.70	1.60	35.80	5.50	25.75	**3.41**
FreGAN (Ours)	**47.85**	**13.49**	**31.05**	**2.44**	**8.97**	**0.91**	**33.39**	**3.76**	**24.93**	3.89

Table 2: The FID (lower is better) and KID (lower is better) scores of our method compared to state-of-the-art methods on $\mathbf{512 \times 512}$ **datasets with limited data amounts**.

	AnimeFace 120 imgs		ArtPainting 1000 imgs		Moongate 136 imgs		Flat 36 imgs		Fauvism 124 imgs	
Method	FID	KID	FID	KID	FID	KID	FID	KID	FID	KID
StyleGAN2 [20]	183.44	242.83	100.35	113.75	288.25	93.14	285.61	214.47	299.15	220.14
SWAGAN [9]	189.71	216.39	56.95	22.50	302.72	99.47	293.94	232.53	291.66	226.21
ADA [17]	59.67	16.02	46.38	12.26	149.06	43.21	248.46	62.89	201.99	86.64
APA [15]	58.38	15.73	47.23	10.60	193.67	50.52	233.52	166.53	197.47	66.13
DiffAug [54]	135.85	148.51	49.25	18.42	136.12	48.04	340.14	247.41	223.58	117.10
FastGAN [25]	55.87	11.17	45.06	10.26	114.79	23.57	216.27	36.88	178.42	58.01
FreGAN (Ours)	**50.19**	**4.58**	**43.13**	**9.71**	**107.13**	**15.58**	**178.10**	**18.35**	**171.95**	**49.81**

Table 3: The FID (lower is better) and KID (lower is better) scores of our method compared to state-of-the-art methods on $\mathbf{1024 \times 1024}$ **datasets with limited data amounts**.

	Shells 64 imgs		Skulls 97 imgs		Pokemon 833 imgs		BrecaHAD 162 imgs		MetFace 1336 imgs	
Method	FID	KID	FID	KID	FID	KID	FID	KID	FID	KID
StyleGAN2 [20]	133.31	33.36	234.54	209.22	161.28	161.98	174.07	176.32	66.97	55.53
SWAGAN [9]	185.96	85.25	203.49	178.96	80.94	68.02	162.53	119.64	31.56	13.96
ADA [17]	133.22	29.12	97.05	12.33	66.41	-	76.67	21.38	24.74	10.23
APA [15]	136.52	58.77	99.46	12.74	51.05	59.29	75.89	25.08	26.03	**5.58**
DiffAug [54]	151.94	54.73	124.23	38.12	62.73	50.68	93.71	31.62	27.45	11.55
FastGAN [25]	141.71	37.00	101.94	12.10	44.96	17.31	59.80	7.24	26.80	7.08
FreGAN (Ours)	**125.77**	**20.58**	**86.12**	**5.47**	**38.88**	**10.42**	**54.88**	**3.41**	**25.42**	5.93

results that, although evaluated on various datasets that have different resolutions and data amounts, our proposed FreGAN achieves superior performance on all these datasets. Our FreGAN consistently improves both FID and KID metrics on 14 of the 15 datasets, demonstrating the effectiveness and generalizability of our proposed techniques. Notably, for those datasets with extremely limited data (less than 100), *i.e.*, Flat (Tab. 2), Shells and Skulls (Tab. 3), our method improves the performance more significantly, *e.g.*, the FID from 216.27 to **178.10** on Flat and from 101.94 to **86.12** on Skulls, and the corresponding KID is improved doubled, further reflecting our model's potential for training GANs with extremely limited data. More quantitative results are presented in the appendix.

Qualitative Comparison. The qualitative results of FastGAN and our FreGAN on various datasets are illustrated in Fig. 4. For each dataset in Fig. 4, from left to right are generated images, the visualization of the low and high-frequency components of the generated images. The images generated by FastGAN contain unsatisfactory artifacts and some of them are incongruous, *e.g.*, the generated images of cat and dog in the bottom right of Fig.4, the cat has artifacts around the head, and the dog's ears are distorted. Our FreGAN significantly facilitates image quality in coordination, rationality, and fine details. As can been seen from Fig. 4, the human face Obama generated by our FreGAN is more photorealistic, the details of the anime face, such as eye color and hair texture, are more realistic, and the synthesized animal faces of cats and dogs are also more plausible. Besides, the frequency components of the images generated by our FreGAN contain wealthier details. For example, the generated image of AnimalFace-Cat has a richer background, and the generated image of Skulls has more clear contours of the eye and nose. Such observation reflects that the proposed FreGAN: 1) ameliorates the quality of generated images under limited data; 2) raises frequency awareness of synthesizing high-frequency signals with richer fine details of images; and 3) takes full advantage of limited data's frequency information. More qualitative results are given in the appendix.

Table 4: The FID (lower is better) and KID (lower is better) scores of our method compared to the state-of-the-art FastGAN on **AFHQ [5] datasets with more training data (∼5k)**.

Method	AFHQ-Cat (5153 imgs)		AFHQ-Dog (4739 imgs)		AFHQ-Wild (4738 imgs)	
	FID	KID	FID	KID	FID	KID
FastGAN [25]	10.17	4.91	25.36	14.29	7.30	1.93
+Ours	**6.62**	**1.95**	**20.75**	**11.45**	**6.37**	**1.31**

Table 5: **Ablation studies on different components of our FreGAN.** We remove each component to evaluate the efficacy of the three ingredients of our method, *i.e.*, HFD, HFA, and FSC. The "Full" represents the the full version combining all three techniques used in the main experiments.

Module	100-shot-Obama (256 × 256)		Anime Face (512 × 512)		Pokemon (1024 × 1024)	
	FID	KID	FID	KID	FID	KID
Baseline	35.80	5.50	55.87	11.17	44.96	17.31
w/o HFD	35.67	7.78	55.17	8.16	41.75	13.69
w/o HFA	34.28	4.60	54.40	10.70	40.27	12.53
w/o FSC	33.52	4.18	51.15	4.83	39.41	11.13
Full	**33.39**	**3.76**	**50.19**	**4.58**	**38.88**	**10.42**

Effectiveness under datasets with more data. To investigate the effectiveness of our FreGAN more comprehensively, we evaluate the performance on datasets with more training data, *i.e.*, AnimalFace-HQ (AFHQ) [5], which includes 3 sub-datasets with close to 5k images, the results are shown in Tab. 4. Similarly, our method yields compelling improvements on both FID and KID metrics when training with more data. Combined with the generated images in Fig. 4, the results further validate our FreGAN's contribution to the synthesize quality. Our method boosts the performance under different amounts of data, suggesting the generalization of our model.

4.2 Ablation Studies

Ablation studies on variants of FreGAN. There are three ingredients of our FreGAN, *i.e.*, the high-frequency discriminator (HFD), high-frequency alignments (HFA), and frequency skip connection (FSC). We evaluate the efficacy of each component by removing each of them from the full version of our FreGAN. We choose one from each of the different resolution datasets, *i.e.*, 100-shot-Obama, Anime face and pokemon for 256, 512 and 1024 resolution, respectively. As shown in Tab. 5, removing any of the three techniques leads to a performance drop, reflecting the contribution of each component. Still, all these variants outperform baseline FastGAN on both FID and KID, which implies that the combination of different components of our method consistently boosts model performance. Moreover, the performance drops the most when removing the HFD module, which is reasonable because the HFD raises the frequency awareness of G and D, and the frequency awareness of D serves as a self-supervision to guide G to synthesize adequate and reasonable frequency signals. Qualitative comparison results of ablation studies are given in the appendix.

Ablation studies on different scale of features. We employ our proposed HFD and HFA on multi-scale features of G and D, namely, 8, 16, and 32 scales of features. Here we provide the ablation studies on different scales in Tab. 6. It can be seen that performing HFD and HFA on multi-scale features boosts the model performance. Besides, when only performing HFD and HFA on single-scale features, the obtained results still outperform the FastGAN baseline, suggesting the effectiveness of HFA and HFD. Notably, despite adding more scales of features may bring further performance advancement, the required additional convolutional and downsampling layers increases for higher scales features(*e.g.*, 128, 256), bringing non-negligible computational costs.

Ablation studies on different frequency components. Three high-frequency components are obtained from wavelet transformation on the features, *i.e.*, LH, HL, and HH. Each of them encodes different details of features as shown in Fig. 3, we sum them to fuse all the detail information for further operation in our main experiments. Here we conduct experiments on the three components respectively to verify their contribution and the necessity of fusing them. As shown in Tab. 6, each high-frequency component contributes to the model performance compared with the baseline, and fusing them can better promote the generation quality.

Table 6: **Ablation studies on the different scales of HFD and HFA on AnimalFace-Dog dataset.** The numbers after "Feat" indicate the scale of features. "LH" and "HL" denote the high-frequency components. The "Full" represents the full version of our FreGAN used in the main experiments.

Module	Metric	Baseline	Feat8	Feat8 + 16	LH	LH + HL	Full
HFD	FID	52.46	51.86	50.63	52.15	51.79	**47.85**
HFD	KID	18.22	17.13	16.28	17.94	16.57	**13.49**
HFA	FID	52.46	52.23	51.93	51.60	49.71	**47.85**
HFA	KID	18.22	17.15	16.87	16.92	15.87	**13.49**

Table 7: The FID (lower is better) and KID (lower is better) scores of our method **combined with model regularization and attention mechanism techniques on 256×256 datasets.**

	Animal Face				100-shot					
	Dog (389 imgs)		Cat (160 imgs)		Panda		Obama		Grumpy_cat	
Method	FID	KID	FID	KID	FID	KID	FID	KID	FID	KID
Lecam [40]	54.88	-	34.18	-	10.16	-	33.16	-	24.93	-
+ Ours	**48.29**	**14.16**	**31.77**	**2.22**	**8.87**	**1.06**	**32.69**	**4.99**	**24.39**	**2.36**
MoCA [23]	54.04	19.25	38.04	8.40	11.24	4.00	42.26	17.03	25.59	4.20
+ Ours	**50.96**	**16.06**	**35.47**	**4.92**	**9.05**	**1.13**	**34.13**	**5.53**	**24.78**	**3.11**

4.3 Analysis on Compatibility and GAN Equilibrium

Compatibility of Our Model. Lecam [40] and MoCA [23] exploit regularization and attention mechanism for training GANs under limited data, respectively. We implement our proposed techniques on them to test the compatibility of our method. We keep the original setting unchanged and the set the regularization weight to 0.1. The FID results are given in Tab. 7, from which we can see that FreGAN can further boost the performance of MoCA and Lecam, demonstrating that our method is complementary to the model regularization and attention mechanism methods.

GAN Equilibrium is improved. Our HFA module aligns the frequency components of real and generated images, guiding G to synthesize precise instead of arbitrary high-frequency signals. Meanwhile, as a byproduct, the HFA mitigates the domain gap between G and D, alleviating the unhealthy competition. As shown in Fig. 5 (a), our discriminator converges to a better point, and our generator can better fool the discriminator, while the discriminator of FastGAN surpass the generator, thus providing less informative guidelines and degrading the synthesize quality. Besides, we plot the FID and KID curves throughout the training process in Fig. 5 (b), from which we can observe that our FreGAN are consistently better. Moreover, we plot the multi-scale HFA loss curves in Fig. 5 (c), where each line denotes the loss of each scale. These curves indicate that the frequency signals are well aligned, lessening the domain gaps and promoting the GAN equilibrium.

5 Discussion

Conclusions. In this paper, we propose a frequency-aware method for training GANs under limited data, *i.e.*, FreGAN. The proposed FreGAN ameliorates the synthesize quality by raising the model's frequency awareness, encouraging the model to pay more attention to frequency signals, especially high-frequency signals, which encode fine details of images. We conduct extensive experiments on various datasets with different amounts of data and different resolutions to demonstrate the efficacy of our proposed method. Qualitative results suggest that our model successfully makes the generator to generate precise high-frequency signals, facilitating high-quality image generation. Quantitative results indicate that our method 1) substantially boosts the performance, especially when data is

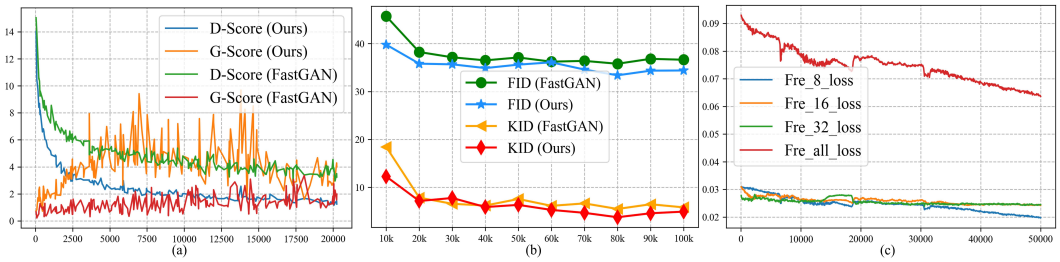


Figure 5: **Equilibrium is improved with our proposed techniques.** (a) Our generator can better deceive the discriminator. (b) Our FreGAN yields better performance throughout the training process. (c) The aligned HFA loss mitigates the asymmetrical information and facilitates the GAN equilibrium.

extremely limited (less than 100), and 2) is complementary to existing regularization and attention models. Moreover, the proposed model alleviates the disequilibrium of GANs by lessening the frequency information gap. In the future, we plan to implement our techniques on more backbones, *e.g.*, StyleGAN2 [20] and apply our method to more applications.

Limitations. Despite achieving significant improvements on various low-data datasets, our FreGAN still struggles in generating photorealistic images when given datasets with limited data but various contents, *e.g.*, only dozens of images, and their contents vary widely. When the low-data datasets are imbalanced [46] or even long-tailed, the proposed method may fail to generalize, which is limited by the intrinsic reasons of the data distribution. Developing more effective ways to train generative models with insufficient training data still requires more efforts.

Acknowledgment

This work is supported by Shanghai Science and Technology Program "Distributed and generative few-shot algorithm and theory research" under Grant No. 20511100600 and "Federated based cross-domain and cross-task incremental learning" under Grant No. 21511100800, Natural Science Foundation of China under Grant No. 62076094, Chinese Defense Program of Science and Technology under Grant No.2021-JCJQ-JJ-0041, China Aerospace Science and Technology Corporation Industry-University-Research Cooperation Foundation of the Eighth Research Institute under Grant No.SAST2021-007.

References

[1] Alper Aksac, Douglas J. Demetrick, Tansel Ozyer, and Reda Alhajj. Brecahad: a dataset for breast cancer histopathological annotation and diagnosis. *BMC Research Notes*, 2019.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[3] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations*, 2018.

[4] Tai-Yin Chiu and Danna Gurari. Photowct2: Compact autoencoder for photorealistic style transfer resulting from blockwise training and skip connections of high-frequency residuals. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2868–2877, 2022.

[5] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020.

[6] Ingrid Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE transactions on information theory*, 36(5):961–1005, 1990.

[7] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[8] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7890–7899, 2020.

[9] Rinon Gal, Dana Cohen Hochberg, Amit Bermano, and Daniel Cohen-Or. Swagan: A style-based wavelet-driven generative model. *ACM Transactions on Graphics (TOG)*, 40(4):1–11, 2021.

[10] Yue Gao, Fangyun Wei, Jianmin Bao, Shuyang Gu, Dong Chen, Fang Wen, and Zhouhui Lian. High-fidelity and arbitrary face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16115–16124, 2021.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[14] Abdul Jabbar, Xi Li, and Bourahla Omar. A survey on generative adversarial networks: Variants, applications, and training. *ACM Computing Surveys*, 54(8):1–49, 2021.

[15] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Deceive d: Adaptive pseudo augmentation for gan training with limited data. *Advances in Neural Information Processing Systems*, 34, 2021.

[16] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[17] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33, 2020.

[18] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[21] Nupur Kumari, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Ensembling off-the-shelf models for gan training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10651–10662, 2022.

[22] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

[23] Tianqin Li, Zijie Li, Andrew Luo, Harold Rockwell, Amir Barati Farimani, and Tai Sing Lee. Prototype memory and attention mechanisms for few shot image generation. In *International Conference on Learning Representations*, 2022.

[24] Yijun Li, Richard Zhang, Jingwan Cynthia Lu, and Eli Shechtman. Few-shot image generation with elastic weight consolidation. *Advances in Neural Information Processing Systems*, 33, 2020.

[25] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards faster and stabilized gan training for high-fidelity few-shot image synthesis. In *International Conference on Learning Representations*, 2021.

[26] Hongyu Liu, Ziyu Wan, Wei Huang, Yibing Song, Xintong Han, and Jing Liao. Pd-gan: Probabilistic diverse gan for image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9371–9381, 2021.

[27] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[28] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*, pages 7176–7185. PMLR, 2020.

[29] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29, 2016.

[30] Utkarsh Ojha, Yijun Li, Jingwan Lu, Alexei A Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10743–10752, 2021.

[31] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

[32] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[33] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *Advances in Neural Information Processing Systems*, 34:17480–17492, 2021.

[34] Katja Schwarz, Yiyi Liao, and Andreas Geiger. On the frequency bias of generative models. *Advances in Neural Information Processing Systems*, 34, 2021.

[35] Zhangzhang Si and Song-Chun Zhu. Learning hybrid image templates (hit) by information projection. *IEEE Transactions on pattern analysis and machine intelligence*, 34(7):1354–1367, 2011.

[36] Vadim Sushko, Dan Zhang, Juergen Gall, and Anna Khoreva. Generating novel scene compositions from single images and videos. *arXiv preprint arXiv:2103.13389*, 2021.

[37] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33, 2020.

[38] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics*, 40(4):1–14, 2021.

[39] Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Trung-Kien Nguyen, and Ngai-Man Cheung. On data augmentation for gan training. *IEEE Transactions on Image Processing*, 30:1882–1897, 2021.

[40] Hung-Yu Tseng, Lu Jiang, Ce Liu, Ming-Hsuan Yang, and Weilong Yang. Regularizing generative adversarial networks under limited data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7921–7931, 2021.

[41] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018.

[42] Lin Wang, Yo-Sung Ho, and Kuk-Jin Yoon. Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10081–10090, 2019.

[43] Yaxing Wang, Abel Gonzalez-Garcia, David Berga, Luis Herranz, Fahad Shahbaz Khan, and Joost van de Weijer. Minegan: effective knowledge transfer from gans to target domains with few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9332–9341, 2020.

[44] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In *Proceedings of the European Conference on Computer Vision*, pages 218–234, 2018.

[45] Yuntao Wang, Lin Zhao, Liman Liu, Huaifei Hu, and Wenbing Tao. Urnet: A u-shaped residual network for lightweight image super-resolution. *Remote Sensing*, 13(19):3848, 2021.

[46] Zhe Wang, Chenjie Cao, and Yujin Zhu. Entropy and confidence-based undersampling boosting random forests for imbalanced problems. *IEEE Transactions on Neural Networks and Learning Systems*, 31(12):5178–5191, 2020.

[47] Zhengwei Wang, Qi She, and Tomas E Ward. Generative adversarial networks in computer vision: A survey and taxonomy. *ACM Computing Surveys (CSUR)*, 54(2):1–38, 2021.

[48] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

[49] Ceyuan Yang, Yujun Shen, Yinghao Xu, and Bolei Zhou. Data-efficient instance generation from instance discrimination. *Advances in Neural Information Processing Systems*, 34:9378–9390, 2021.

[50] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9036–9045, 2019.

[51] Yingchen Yu, Fangneng Zhan, Shijian Lu, Jianxiong Pan, Feiying Ma, Xuansong Xie, and Chunyan Miao. Wavefill: A wavelet-based generation network for image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14114–14123, 2021.

[52] Linfeng Zhang, Xin Chen, Xiaobing Tu, Pengfei Wan, Ning Xu, and Kaisheng Ma. Wavelet knowledge distillation: Towards efficient image-to-image translation. *arXiv preprint arXiv:2203.06321*, 2022.

[53] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[54] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *Advances in Neural Information Processing Systems*, 33, 2020.

[55] Zhengli Zhao, Zizhao Zhang, Ting Chen, Sameer Singh, and Han Zhang. Image augmentations for gan training. *arXiv preprint arXiv:2006.02595*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Section 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code is provided in the supplemental material. We will also make our code and pre-trained models publicly available [3].
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Implement details in Section 4.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Implement details in Section 4.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Section 4.
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] Datasets in Section 4.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] The data we use contains no personally identifiable information or offensive content, the description of datasets is given in Section 4.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

[3] Our codes are available at `https://github.com/kobeshegu/FreGAN_NeurIPS2022`.

A Appendix

In this appendix, we first provide the broader impact of our method. Then we provide more implement details (Sec. A.1), quantitative results of other metrics (Sec. A.2 and Sec. A.3), and qualitative results of generated images and ablation studies (Sec. B.1) that is not elaborated in the main paper. We also provide the quantitative comparison results of our FreGAN and FastGAN on large-scale datasets in Sec.refsec:largedatasets Besides, we provide the latent space interpolation and nearest neighbors found from training images in Sec. C for a more comprehensive analysis on the diversity of our generated images. Lastly, more quantitative and qualitative comparison results of spectral properties are given in Sec. D, demonstrating that our FreGAN is frequency-aware and can indeed produce realistic frequency signals.

Broader impact. The proposed method enables data-efficient GANs training for high-quality image synthesize with limited data. It benefits the practical implementation if GANs in various applications that without sufficient training samples, *e.g.*, medical images and art paintings. And our analysis of the effectiveness of frequency components in image synthesis may also extend the breadth and potential of approaches for training effective GANs from the frequency domain perspective. Besides, being capable of generating plausible and photorealistic images, our method bring potential issue of image abuse and fraud with the generated fake images. However, we believe that the rational use of such advanced technology can bring benefits to more fields like films and art production.

A.1 More implement details.

Implement details of our FreGAN. We perform Haar wavelet transformation on the intermediate 8×8, 16×16, 32×32 features of G and D. The PyTorch-like pseudocode of Haar wavelet transformation is given in Algorithm. 1. We perform FSC by wavelet unpooling the decomposed high-frequency components and feeding the reconstructed features to the subsequent layers. For HFD, we aggregate the high-frequency components by adding LH, HL, HH and then employ additional downsampling and convolutional layers to compute the output scores. Specifically, the architecture resembles the original discriminator. The added layers on the high-frequency components include 2d convolutional, 2d Batch Normalization, and LeakyReLU layers. The difference is that the HFD discriminates the input images' frequency information, raising the discriminator's frequency awareness. For HFA, we align the summed frequency signals of real and generated intermediate features by minimizing Eq.4 in the main paper. Notably, only 1-2 additional layers are added for each high-frequency component without requiring much computational cost. We train our model for 100k iterations and save the checkpoints every 10k iterations. The saved checkpoints are used to generate images for evaluation. All experiments are run on 2 Tesla V100 GPUs with PyTorch framework, and our code will be made available online.

Implement details of baseline methods. We reimplement all the baseline methods with their official code for fair comparisons. For StyleGAN2 [20][4], ADA [17][5], DiffAug [54][6], we keep most of the details unchanged, including style mixing regularization, path length regularization, exponential moving average of weights, non-saturating logistic loss with R_1 regularization. We show the discriminator for 2,000 kimg and use the best training snapshots of each model to generate 5k images for evaluation. For FastGAN [25][7], similarly, we keep all the details unchanged and implement our proposed techniques upon it, we train both our FreGAN and FastGAN for 100k iterations, and we use the saved checkpoints to generate 5k images for evaluation. All of our experiments are run on 2 Tesla V100 GPUs, using PyTorch 1.8.0, and CUDA 11.1. When combining our proposed techniques upon Lecam [40] and MoCA [23], we consistently keep all the details unchanged and add our proposed techniques to them. The coefficient of the regularization term is set as 0.1 following the original paper. Notably, we use the official code and recommended parameter of MoCA. However, we achieve 42.26 on the 100-shot-Obama dataset instead of 37.19 reported in the original paper. We infer this is caused by randomness and hardware differences. Nonetheless, our proposed method contributes to the performance and complements the attention mechanism-based method.

[4] `https://github.com/NVlabs/stylegan2`

[5] `https://github.com/NVlabs/stylegan2-ada`

[6] `https://github.com/mit-han-lab/data-efficient-gans`

[7] `https://github.com/odegeasslbc/FastGAN-pytorch`

Algorithm 1: Haar wavelet transformation pseudocode, PyTorch-like

```
def get_wav(in_channels, pool=True):
    # wavelet decomposition using conv2d
    harr_wav_L = 1 / np.sqrt(2) * np.ones((1, 2))
    harr_wav_H = 1 / np.sqrt(2) * np.ones((1, 2))
    harr_wav_H[0, 0] = -1 * harr_wav_H[0, 0]

    harr_wav_LL = np.transpose(harr_wav_L) * harr_wav_L
    harr_wav_LH = np.transpose(harr_wav_L) * harr_wav_H
    harr_wav_HL = np.transpose(harr_wav_H) * harr_wav_L
    harr_wav_HH = np.transpose(harr_wav_H) * harr_wav_H

    filter_LL = torch.from_numpy(harr_wav_LL).unsqueeze(0)
    filter_LH = torch.from_numpy(harr_wav_LH).unsqueeze(0)
    filter_HL = torch.from_numpy(harr_wav_HL).unsqueeze(0)
    filter_HH = torch.from_numpy(harr_wav_HH).unsqueeze(0)

    if pool:
        net = nn.Conv2d
    else:
        net = nn.ConvTranspose2d
    LL = net(in_channels, in_channels*2,
             kernel_size=2, stride=2, padding=0, bias=False,
             groups=in_channels)
    LH = net(in_channels, in_channels*2,
             kernel_size=2, stride=2, padding=0, bias=False,
             groups=in_channels)
    HL = net(in_channels, in_channels*2,
             kernel_size=2, stride=2, padding=0, bias=False,
             groups=in_channels)
    HH = net(in_channels, in_channels*2,
             kernel_size=2, stride=2, padding=0, bias=False,
             groups=in_channels)

    LL.weight.requires_grad = False
    LH.weight.requires_grad = False
    HL.weight.requires_grad = False
    HH.weight.requires_grad = False

    LL.weight.data = filter_LL.float().unsqueeze(0).expand(in_channels*2, -1, -1, -1)
    LH.weight.data = filter_LH.float().unsqueeze(0).expand(in_channels*2, -1, -1, -1)
    HL.weight.data = filter_HL.float().unsqueeze(0).expand(in_channels*2, -1, -1, -1)
    HH.weight.data = filter_HH.float().unsqueeze(0).expand(in_channels*2, -1, -1, -1)

    return LL, LH, HL, HH

class WavePool(nn.Module):
    def __init__(self, in_channels):
        super(WavePool, self).__init__()
        self.LL, self.LH, self.HL, self.HH = get_wav(in_channels)

    def forward(self, x):
        return self.LL(x), self.LH(x), self.HL(x), self.HH(x)

class WaveUnpool(nn.Module):
    def __init__(self, in_channels, option_unpool='cat5'):
        super(WaveUnpool, self).__init__()
        self.in_channels = in_channels
        self.option_unpool = option_unpool
        self.LL, self.LH, self.HL, self.HH = get_wav(self.in_channels, pool=False)

    def forward(self, LL, LH, HL, HH, original=None):
        if self.option_unpool == 'sum':
            return self.LL(LL) + self.LH(LH) + self.HL(HL) + self.HH(HH)
        elif self.option_unpool == 'cat5' and original is not None:
            return torch.cat([self.LL(LL), self.LH(LH), self.HL(HL), self.HH(HH), original], dim=1)
        else:
            raise NotImplementedError
```

Datasets description.

- AFHQ dataset[8] dataset contain ∼5k training images of animal faces with 512 × resolution. The dataset is made available under the Creative Commons BY-NC 4.0 license.

[8] https://github.com/clovaai/stargan-v2

Table 8: The FID (lower is better) and KID (lower is better) scores of our method compared to state-of-the-art methods on $\mathbf{256 \times 256}$ **datasets with limited data amounts**.

Method	Medici (100 imgs)		Temple (100 imgs)		Bridge (100 imgs)		Wuzhen (100 imgs)	
	FID	KID	FID	KID	FID	KID	FID	KID
StyleGAN2 [20]	66.36	41.16	73.35	46.76	116.40	71.51	135.40	116.46
ADA [17]	44.21	-	49.72	-	72.07	-	92.81	47.01
APA [15]	76.11	-	41.38	10.33	189.74	98.45	102.10	45.92
DiffAug [54]	42.63	21.23	50.73	11.19	49.97	11.92	122.44	78.08
FastGAN [25]	38.47	12.63	36.01	4.95	46.82	8.25	67.98	15.13
FreGAN (Ours)	**27.30**	**3.34**	**33.38**	**3.23**	**44.18**	**7.25**	**59.89**	**6.62**

- 100-shot datasets[9] contain various contents of images, and all the datasets contain 100 training images. They are ideal for verifying the quality of the generation in low-shot scenarios.
- MetFace[10] dataset contains 1336 high-quality PNG images at 1024×1024 resolution. The dataset is made available under the Creative Commons BY-NC 2.0 license.
- BrecaHAD[11] dataset contains 162 images for breast cancer histopathological annotation and diagnosis. Its texture and content are complex, thus is suited for evaluating GANs' performance under limited data, facilitating the exploration of data-efficient GANs for downstream tasks of the medical field.
- anime-face, art-paintings, moongate, flat, fauvism, shells, skulls[12]. These datasets include 60 1000 images with different resolutions. Thus we adopt them for evaluating our model under limited data. We resize them to the closest resolution in implementation.

All of the datasets we used in this paper are open-sourced, and we use them only for academic research without any commercial purposes.

A.2 More Quantitative comparison on datasets with limited data amounts.

We evaluate the performance of our FreGAN and baseline models on more datasets with limited data amounts in Tab. 8, namely, Medici, Temple, Bridge, and Wuzhen, all of which contain only 100 training images. The resolution of these datasets is 256×256. The FID and KID results are consistent with the results in the main paper. Our FreGAN achieves better performance compared with the baseline models, suggesting the effectiveness and generalization of our model.

A.3 More Quantitative Comparison on Datasets with limited data amounts.

In addition to the two common-used metrics, *i.e.*, FID and KID, we also compute the Precision, Recall [22], Density, Coverage [28], Inception Score (IS) [32], and LPIPS [53] results in Tab. 9, Tab. 10, Tab. 11, Tab. 12, Tab. 13, Tab. 14, Tab. 15, Tab. 16, Tab. 17, Tab. 18, Tab. 19, Tab. 20, Tab. 21, Tab. 22, Tab. 23, Tab. 24, Tab. 25, Tab. 26, Tab. 27.

Precision is quantified by calculating if the generated images are within the estimated manifold of real images, and symmetrically, Recall is quantified by calculating if the real images are within the estimated manifold of generated images [22]. Precision evaluates the probability of generated images falling into the real distribution, and Recall is the opposite. Density and Coverage evaluate the fidelity and diversity of generative models, respectively [28], they are claimed successfully detect two identical distributions and are not robust against outliers. Specifically, we use the generated 5k images of each model corresponding to the best FID results. The whole training images are used as the referenced distribution. We set the nearest k as 5 and compute the Precision, Recall, Density, and Coverage based on the official code of [28][13]. We split the generated into 10 parts for IS and

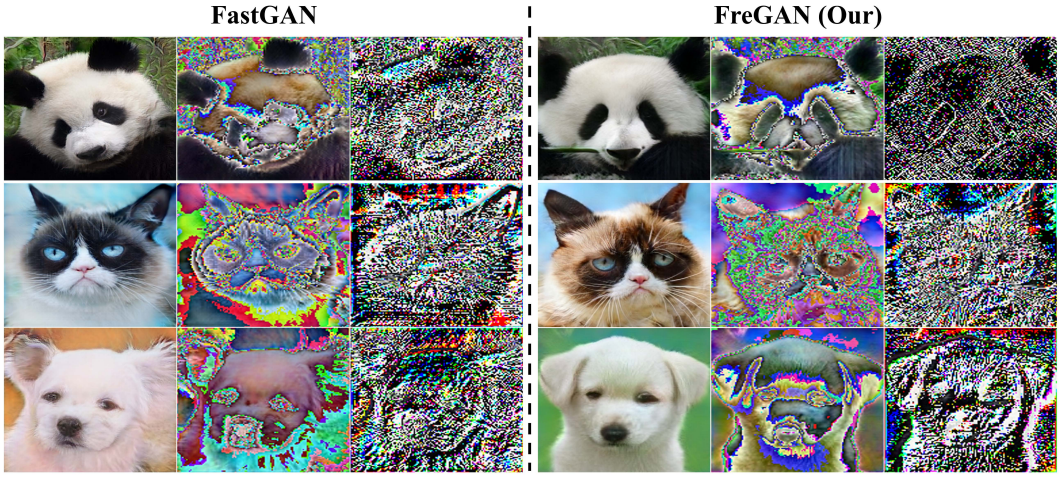
[9] https://data-efficient-gans.mit.edu/datasets/

[10] https://github.com/NVlabs/metfaces-dataset

[11] https://figshare.com/articles/dataset/BreCaHAD_A_Dataset_for_Breast_Cancer_Histopathological_Annotation_and_Diagnosis/7379186

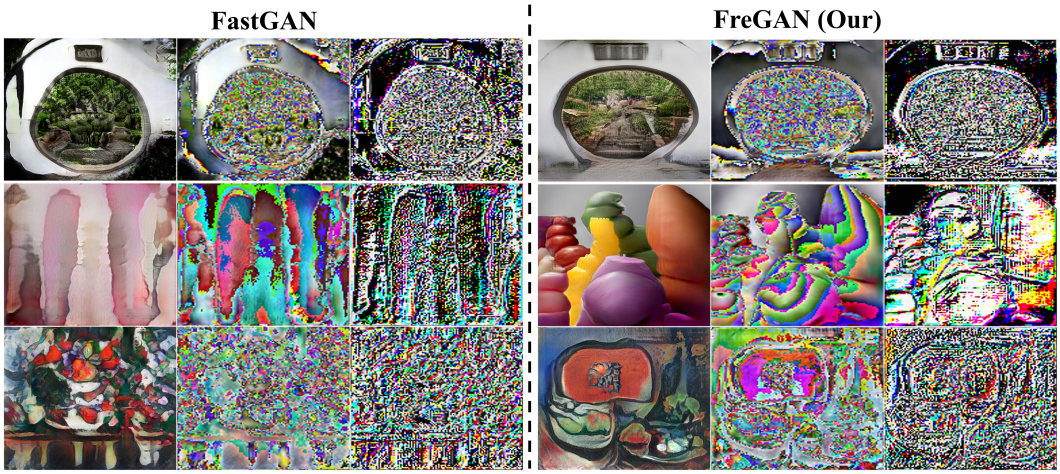
[12] https://drive.google.com/file/d/1aAJCZbXNHyraJ6Mi13dSbe7pTyfPXha0/view/

[13] https://github.com/clovaai/generative-evaluation-prdc



256*256: Panda, Grumpy_cat (100 images), AnimalFace-Dog (389 images)

Figure 6: **Qualitative comparison results of our FreGAN and baseline FastGAN.** The images from left to right are generated images, low-frequency, and high-frequency components, respectively. Our FreGAN improves the overall quality of generated images and raises the model's frequency awareness, encouraging the generator to produce precise high-frequency signals with fine details.



512*512: Moongate (136 images), Flat (36 images), Fauvism (124 images)

Figure 7: **Qualitative comparison results of our FreGAN and baseline FastGAN.** The images from left to right are generated images, low-frequency, and high-frequency components, respectively. Our FreGAN improves the overall quality of generated images and raises the model's frequency awareness, encouraging the generator to produce precise high-frequency signals with fine details.

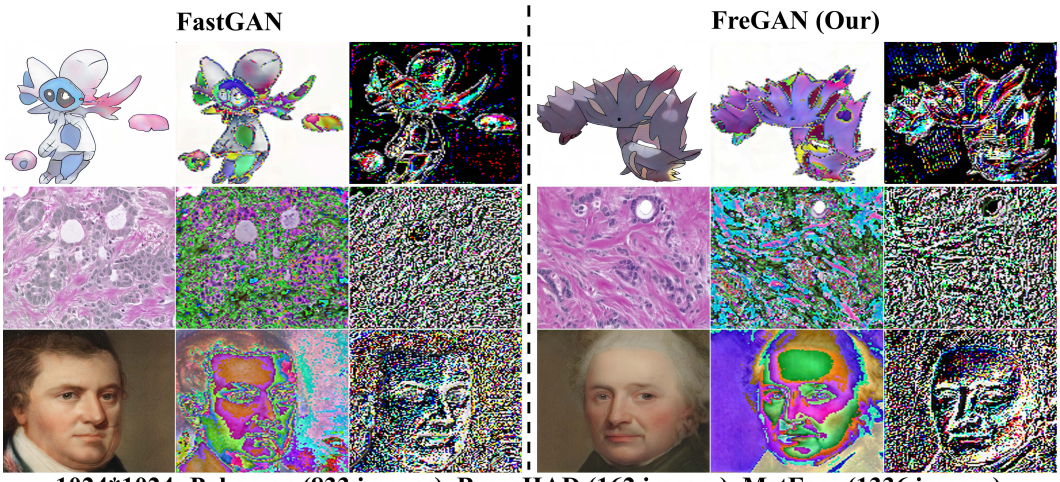


Figure 8: **Qualitative comparison results of our FreGAN and baseline FastGAN.** The images from left to right are generated images, low-frequency, and high-frequency components, respectively. Our FreGAN improves the overall quality of generated images and raises the model's frequency awareness, encouraging the generator to produce precise high-frequency signals with fine details.

Table 9: The Precision (P) (higher is better) and Recall (R) (higher is better) scores of our method compared to state-of-the-art methods on $\mathbf{256 \times 256}$ **datasets with limited data amounts**.

	Animal Face				100-shot					
	Dog (389 imgs)		Cat (160 imgs)		Panda		Obama		Grumpy_cat	
Method	P	R	P	R	P	R	P	R	P	R
StyleGAN2 [20]	0.33	0.06	0.40	0.04	0.61	0.06	0.47	**0.16**	0.74	0.10
ADA [17]	0.78	**0.50**	0.78	0.13	0.61	0.06	0.90	0.00	0.72	0.00
APA [15]	0.75	0.08	0.78	0.13	0.48	0.14	0.89	0.00	0.88	0.05
DiffAug [54]	0.79	0.31	0.87	1.05	0.79	0.11	0.85	0.00	**0.97**	0.01
FastGAN [25]	**0.88**	**0.50**	0.87	**0.25**	**0.91**	0.10	0.94	0.10	0.93	**0.13**
FreGAN (Ours)	0.86	0.38	**0.90**	0.24	**0.91**	**0.15**	**0.96**	0.11	0.94	0.03

report the mean and standard deviation of the calculated IS scores. We compute LPIPS [53][14] of all the paired of 5k images generated by each methods and report the average and standard deviation of LPIPS scores [36].

We can observe from these tables that: 1) KID and FID can consistently reflect the synthesize quality under limited data; 2) The Precision and Recall serve as supplementary metrics of evaluating generative GANs. They can reflect the distance between the generated and real distributions. However, they may be biased when training data is limited since the model tends to overfit when given limited data, the model simply replicates the training images and achieves high Precision and Recall; 3) Evaluating the fidelity of generated images, the Density metric is more consistent with FID and KID, while the Coverage is not suited for evaluating the diversity as the synthesized 5k images cover the original training images, leading high value of the Coverage, *e.g.*, equals to 1 on many datasets and baseline models; 4) Despite may be biased, our proposed FreGAN achieves advanced performance on most of the used datasets and the adopted evaluation metrics, demonstrating the effectiveness and superiority of our method; 5) Notably, the recall is very low for all datasets and methods. We infer that this is possibly due to the low diversity of the training and the fact that recall is not suitable for evaluation in scenarios with limited data; 6) Devising indicative evaluation metrics for few-shot image generation tasks remains an open and tricky problem because one has to consider the degree of overfitting, the fidelity, and the diversity.

[14] `https://github.com/richzhang/PerceptualSimilarity`

Table 10: The Precision (P) (higher is better) and Recall (R) (higher is better) scores of our method compared to state-of-the-art methods on $\mathbf{256 \times 256}$ **datasets with limited data amounts**.

Method	Medici (100 imgs)		Temple (100 imgs)		Bridge (100 imgs)		Wuzhen (100 imgs)	
	P	R	P	R	P	R	P	R
StyleGAN2 [20]	0.28	0.02	0.47	**0.06**	0.79	0.00	0.09	0.04
ADA [17]	-	-	-	-	-	-	0.50	0.00
APA [15]	-	-	0.91	0.01	0.25	0.00	0.44	0.06
DiffAug [54]	0.60	0.00	0.88	**0.06**	0.89	**0.14**	0.30	0.01
FastGAN [25]	0.87	0.00	0.91	0.05	0.88	0.09	0.80	**0.15**
FreGAN (Ours)	**0.93**	**0.03**	**0.93**	0.02	**0.90**	0.04	**0.85**	0.09

Table 11: The Precision (P) (higher is better) and Recall (R) (higher is better) scores of our method compared to state-of-the-art methods on $\mathbf{512 \times 512}$ **datasets with limited data amounts**.

Method	AnimeFace 120 imgs		ArtPainting 1000 imgs		Moongate 136 imgs		Flat 36 imgs		Fauvism 124 imgs	
	P	R	P	R	P	R	P	R	P	R
StyleGAN2 [20]	0.86	0.00	0.34	0.01	0.63	**0.16**	0.62	0.00	0.35	0.00
ADA [17]	0.89	0.03	0.71	0.35	0.54	0.01	0.75	0.00	0.76	0.00
APA [15]	0.91	0.03	0.67	**0.38**	0.46	0.00	0.50	0.00	0.73	0.00
DiffAug [54]	0.58	0.00	0.74	0.20	0.70	0.00	0.86	0.00	0.48	0.00
FastGAN [25]	0.89	0.12	0.81	0.32	**0.71**	0.02	0.74	**0.03**	**0.84**	**0.02**
FreGAN (Ours)	**0.93**	**0.13**	**0.83**	0.33	**0.71**	0.06	**0.90**	**0.03**	0.82	**0.02**

Table 12: The Precision (P) (higher is better) and Recall (R) (higher is better) scores of our method compared to state-of-the-art methods on $\mathbf{1024 \times 1024}$ **datasets with limited data amounts**.

Method	Shells 64 imgs		Skulls 97 imgs		Pokemon 833 imgs		BrecaHAD 162 imgs		MetFace 1336 imgs	
	P	R	P	R	P	R	P	R	P	R
StyleGAN2 [20]	**0.73**	0.03	0.12	0.02	0.69	0.00	0.53	0.01	0.74	0.00
ADA [17]	0.52	0.03	0.72	0.03	-	-	0.82	0.12	0.78	0.23
APA [15]	0.50	0.02	0.76	0.06	**0.87**	0.00	0.83	0.24	0.80	0.27
DiffAug [54]	0.56	0.00	0.57	0.00	0.69	0.01	0.68	0.04	0.82	0.24
FastGAN [25]	0.59	0.06	0.70	0.03	0.74	0.25	**0.94**	0.42	**0.86**	0.27
FreGAN (Ours)	0.65	**0.08**	**0.83**	**0.08**	0.80	**0.31**	**0.94**	**0.51**	**0.86**	**0.32**

Table 13: The Precision (P) (higher is better) and Recall (R) (higher is better) scores of our method compared to the baseline FastGAN [25] on **AFHQ ($\sim$5k) [5] datasets with more data**.

Method	AFHQ-Cat (5153 imgs)		AFHQ-Dog (4739 imgs)		AFHQ-Wild (4738 imgs)	
	P	R	P	R	P	R
FastGAN [25]	0.81	0.31	0.86	0.56	0.76	**0.22**
+Ours	**0.82**	**0.45**	**0.87**	**0.69**	**0.77**	0.21

Table 14: The Density (D) (higher is better) and Coverage (C) (higher is better) scores of our method compared to state-of-the-art methods on $\mathbf{256 \times 256}$ **datasets with limited data amounts**.

Method	Animal Face				100-shot					
	Dog (389 imgs)		Cat (160 imgs)		Panda		Obama		Grumpy_cat	
	D	C	D	C	D	C	D	C	D	C
StyleGAN2 [20]	0.19	0.51	0.28	0.89	0.06	0.99	0.26	0.98	0.59	0.99
ADA [17]	0.61	0.96	0.85	**1.00**	0.06	**1.00**	1.23	**1.00**	0.30	0.45
APA [15]	0.62	0.74	0.78	0.99	0.14	0.91	0.97	**1.00**	0.95	**1.00**
DiffAug [54]	0.65	0.91	1.13	**1.00**	0.11	**1.00**	0.68	**1.00**	**1.37**	**1.00**
FastGAN [25]	**0.87**	0.96	1.06	**1.00**	0.10	**1.00**	1.28	**1.00**	1.30	**1.00**
FreGAN (Ours)	0.86	**0.98**	**1.24**	**1.00**	**0.15**	**1.00**	**1.38**	**1.00**	1.28	**1.00**

Table 15: The Density (D) (higher is better) and Coverage (C) (higher is better) scores of our method compared to state-of-the-art methods on **256×256 datasets with limited data amounts**.

Method	Medici (100 imgs)		Temple (100 imgs)		Bridge (100 imgs)		Wuzhen (100 imgs)	
	D	C	D	C	D	C	D	C
StyleGAN2 [20]	0.11	0.79	0.31	0.81	0.64	0.55	0.04	0.67
ADA [17]	-	-	-	-	-	-	0.27	0.90
APA [15]	-	-	**1.27**	0.98	0.06	0.23	0.29	0.96
DiffAug [54]	0.35	0.86	0.91	**1.00**	1.03	**1.00**	0.14	0.83
FastGAN [25]	0.88	0.98	1.19	**1.00**	1.04	**1.00**	0.94	**1.00**
FreGAN (Ours)	**1.00**	**1.00**	1.24	**1.00**	**1.16**	**1.00**	**1.20**	**1.00**

Table 16: The Density (D) (higher is better) and Coverage (C) (higher is better) scores of our method compared to state-of-the-art methods on **512×512 datasets with limited data amounts**.

Method	AnimeFace 120 imgs		ArtPainting 1000 imgs		Moongate 136 imgs		Flat 36 imgs		Fauvism 124 imgs	
	D	C	D	C	D	C	D	C	D	C
StyleGAN2 [20]	0.25	0.10	0.16	0.34	0.19	0.38	0.36	0.67	0.16	0.27
ADA [17]	1.58	0.99	0.93	0.91	0.45	0.99	0.90	0.83	0.99	0.90
APA [15]	**2.15**	**1.00**	0.72	0.90	0.42	0.96	0.37	0.67	1.13	0.93
DiffAug [54]	0.22	0.30	0.90	0.87	0.86	0.90	0.93	0.56	0.32	0.75
FastGAN [25]	1.27	**1.00**	1.17	0.95	0.93	**1.00**	0.74	0.97	1.53	0.99
FreGAN (Ours)	1.65	**1.00**	**1.23**	**0.97**	**1.17**	**1.00**	**0.94**	**1.00**	**1.33**	**1.00**

Table 17: The Density (D) (higher is better) and Coverage (C) (higher is better) scores of our method compared to state-of-the-art methods on **1024×1024 datasets with limited data amounts**.

Method	Shells 64 imgs		Skulls 97 imgs		Pokemon 833 imgs		BrecaHAD 162 imgs		MetFace 1336 imgs	
	D	C	D	C	D	C	D	C	D	C
StyleGAN2 [20]	**0.97**	**1.00**	0.11	0.71	0.39	0.13	0.23	0.47	0.61	0.51
ADA [17]	0.43	**1.00**	0.86	**1.00**	-	-	0.75	**1.00**	0.93	0.96
APA [15]	0.35	0.89	0.91	**1.00**	**1.22**	0.71	0.68	**1.00**	1.12	0.96
DiffAug [54]	0.47	0.94	0.58	0.99	0.50	0.63	0.52	0.96	1.08	0.96
FastGAN [25]	0.66	0.94	1.34	**1.00**	0.92	0.96	**1.20**	**1.00**	**1.37**	0.96
FreGAN (Ours)	0.78	**1.00**	**1.35**	**1.00**	1.10	**0.97**	1.08	**1.00**	1.27	**0.97**

Table 18: The Density (D) (higher is better) and Coverage (C) (higher is better) scores of our method compared to the baseline FastGAN [25] on **AFHQ ($\sim$5k) [5] datasets with more data**.

Method	AFHQ-Cat (5153 imgs)		AFHQ-Dog (4739 imgs)		AFHQ-Wild (4738 imgs)	
	D	C	D	C	D	C
FastGAN [25]	**1.19**	0.80	**0.82**	0.50	1.21	0.72
+Ours	1.18	**0.85**	0.72	**0.57**	**1.24**	**0.73**

Table 19: The IS (higher is better) scores of our method compared to state-of-the-art methods on **256×256 datasets with limited data amounts**.

Method	Animal Face		100-shot		
	Dog (389 imgs)	Cat (160 imgs)	Panda	Obama	Grumpy_cat
	IS	IS	IS	IS	IS
StyleGAN2 [20]	7.29±0.33	2.37±0.08	**1.03**±0.01	**1.67**±0.05	1.31±0.02
ADA [17]	8.13±0.30	2.43±0.06	1.01±0.00	1.38±0.03	1.10±0.01
APA [15]	7.35±0.27	2.37±0.08	1.02±0.00	1.45±0.01	**1.43**±0.02
DiffAug [54]	8.22±0.31	2.03±0.05	1.01±0.00	1.29±0.02	1.29±0.01
FastGAN [25]	7.60±0.30	2.28±0.06	1.00±0.00	1.32±0.02	1.33±0.02
FreGAN (Ours)	**8.75**±0.33	**2.47**±0.06	1.00±0.00	1.50±0.02	1.35±0.01

B Results of Large Datasets

We evaluate the effectiveness of our method on large-scale datasets in Tab. 28 and Tab. 29. Specifically, we use the whole datasets of celebA [27] as training data, and we randomly select 30k images from

Table 20: The IS (higher is better) scores of our method compared to state-of-the-art methods on **256×256 datasets with limited data amounts**.

Method	Medici (100 imgs)	Temple (100 imgs)	Bridge (100 imgs)	Wuzhen (100 imgs)
	IS	IS	IS	IS
StyleGAN2 [20]	1.32±0.03	**2.04**±0.06	1.60±0.03	1.93±0.04
ADA [17]	**2.18**±0.04	1.98±0.04	**1.92**±0.03	2.02±0.03
APA [15]	1.24±0.01	1.60±0.02	1.08±0.01	**2.43**±0.07
DiffAug [54]	1.78±0.03	1.76±0.02	1.68±0.03	1.93±0.05
FastGAN [25]	1.76±0.04	1.63±0.02	1.55±0.02	1.99±0.04
FreGAN (Ours)	1.69±0.02	1.62±0.01	1.59±0.03	2.12±0.05

Table 21: The IS (higher is better) scores of our method compared to state-of-the-art methods on **512×512 datasets with limited data amounts**.

Method	AnimeFace 120 imgs	ArtPainting 1000 imgs	Moongate 136 imgs	Flat 36 imgs	Fauvism 124 imgs
	IS	IS	IS	IS	IS
StyleGAN2 [20]	1.37±0.02	2.79±0.08	3.82±0.06	2.21±0.04	2.32±0.04
ADA [17]	1.93±0.04	3.64±0.12	4.12±0.19	3.77±0.09	2.93±0.07
APA [15]	1.85±0.06	**4.65**±0.17	**4.19**±0.17	3.02±0.05	**3.68**±0.10
DiffAug [54]	1.19±0.01	3.39±0.06	2.66±0.07	1.48±0.02	3.14±0.10
FastGAN [25]	2.06±0.04	4.25±0.12	2.96±0.09	4.55±0.14	3.24±0.07
FreGAN (Ours)	**2.08**±0.03	4.17±0.14	3.39±0.07	**5.18**±0.21	3.20±0.10

Table 22: The IS (higher is better) scores of our method compared to state-of-the-art methods on **1024×1024 datasets with limited data amounts**.

Method	Shells 64 imgs	Skulls 97 imgs	Pokemon 833 imgs	BrecaHAD 162 imgs	MetFace 1336 imgs
	IS	IS	IS	IS	IS
StyleGAN2 [20]	3.24±0.10	**3.97**±0.09	2.13±0.04	1.65±0.03	1.92±0.04
ADA [17]	3.78±0.09	2.61±0.08	1.41±0.02	2.82±0.10	3.38±0.09
APA [15]	**3.83**±0.14	2.82±0.09	1.99±0.44	**3.01**±0.13	**3.51**±0.12
DiffAug [54]	3.14±0.10	2.67±0.08	2.60±0.07	2.72±0.10	3.29±0.10
FastGAN [25]	2.52±0.09	2.24±0.08	**2.63**±0.09	2.83±0.04	3.07±0.08
FreGAN (Ours)	2.72±0.06	2.47±0.06	2.38±0.03	**3.01**±0.05	3.06±0.08

Table 23: The IS (higher is better) scores of our method compared to the baseline FastGAN [25] on **AFHQ (∼5k) [5] datasets with more data**.

Method	AFHQ-Cat (5153 imgs)	AFHQ-Dog (4739 imgs)	AFHQ-Wild (4738 imgs)
	IS	IS	IS
FastGAN [25]	1.93±0.02	8.44±0.27	5.08±0.08
+Ours	**2.07**±0.05	**9.23**±0.31	**5.14**±0.10

Table 24: The LPIPS (higher is better) scores of our method compared to state-of-the-art methods on 256×256 datasets with limited data amounts.

Method	Animal Face		100-shot		
	Dog (389 imgs)	Cat (160 imgs)	Panda	Obama	Grumpy_cat
	LPIPS	LPIPS	LPIPS	LPIPS	LPIPS
StyleGAN2 [20]	0.6550±0.0011	0.5774±0.0010	0.4935±0.0013	0.5264±0.0013	0.4632±0.0013
ADA [17]	0.6499±0.0011	0.6145±0.0014	0.4996±0.0012	0.4717±0.0015	0.4721±0.0017
APA [15]	0.6296±0.0013	0.6310±0.0013	0.5037±0.0011	0.4922±0.0012	0.4519±0.0011
DiffAug [54]	0.6301±0.0009	0.5453±0.0013	0.5144±0.0010	0.4727±0.0013	0.4426±0.0013
FastGAN [25]	0.6751±0.0011	0.6452±0.0014	0.6073±0.0009	**0.6081**±0.0012	0.6077±0.0009
FreGAN (Ours)	**0.6848**±0.0011	**0.6671**±0.0014	**0.6089**±0.0010	0.6025±0.0011	**0.6149**±0.0009

FFHQ [19] as training data. We generate 50k images for quantitative evaluation, we can observe from Tab. 28 and Tab. 29 that our FreGAN also improves the generation quality on large-scale training data.

Table 25: The LPIPS (higher is better) scores of our method compared to state-of-the-art methods on 256×256 datasets with limited data amounts.

Method	Medici (100 imgs)	Temple (100 imgs)	Bridge (100 imgs)	Wuzhen (100 imgs)
	LPIPS	LPIPS	LPIPS	LPIPS
StyleGAN2 [20]	0.5410±0.0013	0.5280±0.0017	0.4651±0.0014	**0.6736**±0.0012
ADA [17]	0.5182±0.0030	0.4389±0.0024	0.5341±0.0021	0.6086±0.0015
APA [15]	0.3550±0.0027	0.4909±0.0017	0.5872±0.0013	0.6622±0.0014
DiffAug [54]	0.4683±0.0017	0.5250±0.0014	0.5674±0.0011	0.6724±0.0013
FastGAN [25]	**0.5298**±0.0024	0.5229±0.0015	**0.5958**±0.0010	0.6630±0.0011
FreGAN (Ours)	**0.5298**±0.0024	**0.5275**±0.0015	0.5947±0.0009	0.6656±0.0011

Table 26: The LPIPS (higher is better) scores of our method compared to state-of-the-art methods on 512×512 datasets with limited data amounts.

Method	AnimeFace 120 imgs	ArtPainting 1000 imgs	Moongate 136 imgs	Flat 36 imgs	Fauvism 124 imgs
	LPIPS	LPIPS	LPIPS	LPIPS	LPIPS
StyleGAN2 [20]	0.4253±0.0020	0.7244±0.0009	0.7047±0.0026	0.6223±0.0026	0.6344±0.0009
ADA [17]	0.5611±0.0015	0.8102±0.0015	0.6418±0.0015	0.7288±0.0017	0.6509±0.0014
APA [15]	0.5491±0.0017	0.8062±0.0014	**0.7235**±0.0016	0.6317±0.0022	0.6848±0.0014
DiffAug [54]	0.4926±0.0005	0.7717±0.0016	0.5880±0.0015	0.4403±0.0005	0.6117±0.0023
FastGAN [25]	0.6188±0.0011	0.8344±0.0015	0.6603±0.0010	0.7939±0.0016	**0.7028**±0.0010
FreGAN (Ours)	**0.6191**±0.0010	**0.8439**±0.0016	0.6673±0.0016	**0.7952**±0.0011	**0.7028**±0.0010

Table 27: The LPIPS (higher is better) scores of our method compared to state-of-the-art methods on 1024×1024 datasets with limited data amounts.

Method	Shells 64 imgs	Skulls 97 imgs	Pokemon 833 imgs	BrecaHAD 162 imgs	MetFace 1336 imgs
	LPIPS	LPIPS	LPIPS	LPIPS	LPIPS
StyleGAN2 [20]	**0.5486**±0.0018	**0.6565**±0.0024	0.5870±0.0006	0.4604±0.0016	0.5321±0.0016
ADA [17]	0.5268±0.0015	0.5857±0.0025	0.4050±0.0015	0.5524±0.0016	0.6648±0.0013
APA [15]	0.5337±0.0017	0.6208±0.0025	0.4241±0.0013	0.4874±0.0016	**0.6947**±0.0014
DiffAug [54]	0.4935±0.0015	0.6027±0.0028	0.4811±0.0011	0.4961±0.0024	0.6579±0.0013
FastGAN [25]	0.4908±0.0012	0.6115±0.0028	0.5705±0.0008	**0.5597**±0.0012	0.6698±0.0013
FreGAN (Ours)	0.4957±0.0014	0.6112±0.0026	**0.5770**±0.0008	0.5582±0.0011	0.6706±0.0012

Table 28: Comparison results of our method compared to the baseline FastGAN [25] on **CelebA dataset with 30k training images**.

Method	FID (↓)	KID (↓)	IS	Precision	Recall	Density	Coverage
FastGAN [25]	23.35	19.04	2.57±0.07	0.69	0.25	0.60	0.35
FreGAN (Ours)	**20.65**	**15.73**	**2.58**±0.06	**0.70**	**0.29**	**0.67**	**0.39**

Table 29: Comparison results of our method compared to the baseline FastGAN [25] on **FFHQ dataset with 30k training images**.

Method	FID (↓)	KID (↓)	IS	Precision	Recall	Density	Coverage
FastGAN [25]	26.17	14.77	3.41±0.15	0.67	0.34	0.61	0.42
FreGAN (Ours)	**23.62**	**12.44**	**3.51**±0.10	**0.69**	**0.37**	**0.66**	**0.44**

In the future, we will also combine our proposed method with large-scale GANs like StyleGAN3 [18] and BigGAN [7] for more comprehensive investigation.

B.1 More Qualitative Results

We provide more qualitative results of FastGAN and our FreGAN in Fig. 6, Fig. 7, Fig. 8, respectively. Our FreGAN is capable of generating more realistic images with more fine details. The high-frequency components of our generated images contain more information, such as the eyes and mouth of the Panda in Fig. 6, the texture of Moongate in Fig. 7.

Figure 9: **Qualitative results of our FreGAN.** The left part shows some of real training images and the right part of images are generated by our FreGAN. Our FreGAN is capable of generating photorealistic images with fine details, which is indistinguishable by the discriminator.

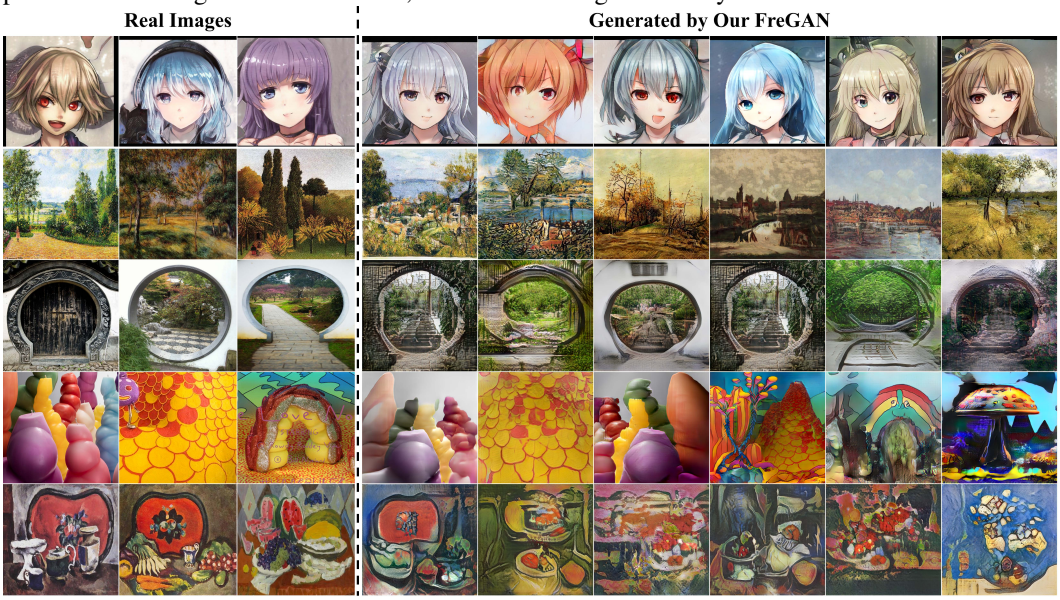


Figure 10: **Qualitative results of our FreGAN.** The left part shows some of real training images and the right part of images are generated by our FreGAN. Our FreGAN is capable of generating photorealistic images with fine details, which is indistinguishable by the discriminator.

We give more generated images of our FreGAN in Fig. 9, Fig. 10, and Fig. 11. We can observe from these figures that our FreGAN can produce vivid images with fine details. The photorealistic generated images indicate the effectiveness of our FreGAN in improving the generation quality of GANs under limited data. However, our FreGAN still struggles in generating photorealistic images when given datasets with limited data but various contents, *e.g.*, only dozens of images, and their contents vary widely. As can be seen from Fig. 11, the overall quality of the generated Pokemon images is not satisfactory because the Pokemon dataset contains multiple categories of Pokemon and only a few images for each character, making it hard to produce high-quality Pokemon images.

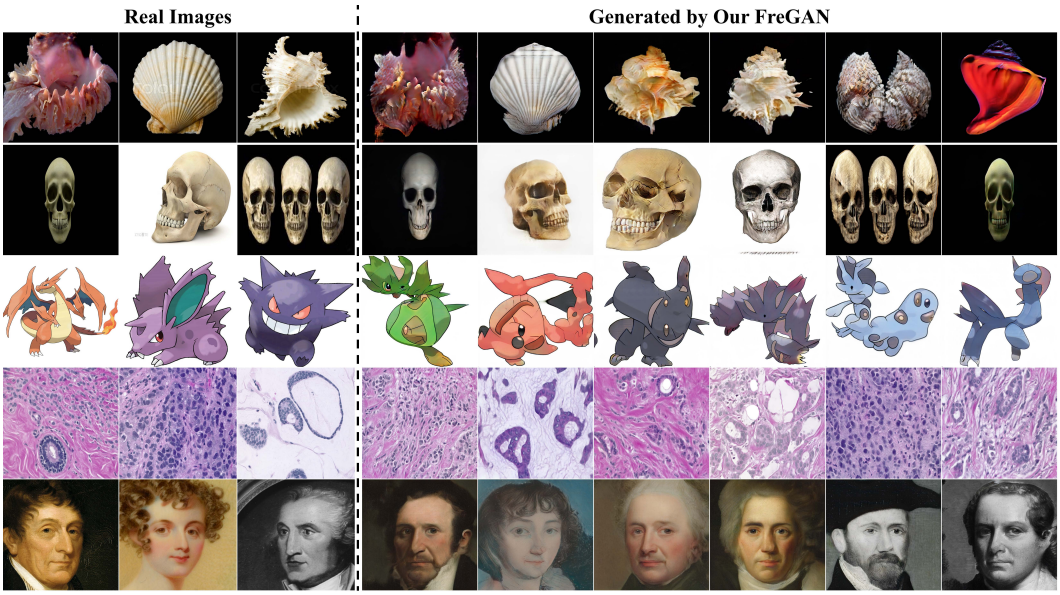


Figure 11: **Qualitative results of our FreGAN.** The left part shows some of real training images and the right part of images are generated by our FreGAN. Our FreGAN is capable of generating photorealistic images with fine details, which is indistinguishable by the discriminator.

Figure 12: **Qualitative comparison results of ablation studies.** Each component of our method contributes to the quality of the generated images. HFD improves the fine details like the eyes of the dog and the color of eyes and hairs of the AnimeFace, and HFA makes nostrils, teeth more realistic.

Figure 13: **Latent space interpolation results of our FreGAN.** The smooth transition images suggest that our FreGAN is capable of generating images instead of memorizing images.

C Analysis on generation diversity

We provide the latent space interpolation results of our FreGAN in Fig. 13, from which we can observe that the transition of images generated by different latent codes are smooth and photo-realistic, indicating that our FreGAN promotes the generation quality without compromising the generation diversity. Moreover, we find the closest real images to the generated ones from training data based on LPIPS score, the visualization results are given in Fig. 14, Fig. 15, and Fig. 16. The results demonstrate that our FreGAN learns to produce new images instead of memorizing training images. For example, the body hair color, perspective, and demeanor of dogs are different. The mouth, eyes, hairstyle of AnimeFace are different. And for the Shells dataset in Fig. 16, different generated images that have the closest distance with the same real image are different in color, shape, etc, which further demonstrating that our method improves generation quality meanwhile maintaining diversity. To investigate the training process of our FreGAN, we plot the outputs of the discriminator throughout the training in Fig. 17, the stable curve of our FreGAN demonstrate that our FreGAN can be trained more effective.

Figure 14: **Nearest real samples to the generated ones on AnimalFace Dog dataset.** For each paired images, the left one is generated by our FreGAN and the right one is the closest image found from the training data. We adopt LPIPS score to measure the similarity of images, *i.e.*, the closest image found from the training data have the largest LPIPS score with the generated one.

Figure 15: **Nearest real samples to the generated ones on AnimeFace dataset.** For each paired images, the left one is generated by our FreGAN and the right one is the closest image found from the training data. We adopt LPIPS score to measure the similarity of images, *i.e.*, the closest image found from the training data have the largest LPIPS score with the generated one.

D More analysis on spectral properties of generated images

We first give 2D DWT results of real images in Fig. 18. Then we provide the 2D DWT visualization results in Fig. 19 and Fig. 20, which complement the qualitative comparison results in Figure.4 of the main paper. We can observe that although presented in different ways of visualization, images generated by our FreGAN contain more realistic frequency signals, indicating the efficacy of our proposed techniques. Besides, we compare the averaged 2D power spectrum, one-dimensional slices of the power spectrum, the power spectrum distance, and the statistic (mean and variance power spectrum) in Fig. 21, Fig. 22, Fig. 23, and Fig. 24, respectively. We can observe from these figures that: 1) Our FreGAN can produce more realistic frequency signals compared with other methods; 2) The overlap between the generated images of our FreGAN and the training data is the largest

Figure 16: **Nearest real samples to the generated ones on Shells dataset.** For each paired images, the left one is generated by our FreGAN and the right one is the closest image found from the training data. We adopt LPIPS score to measure the similarity of images, *i.e.*, the closest image found from the training data have the largest LPIPS score with the generated one.

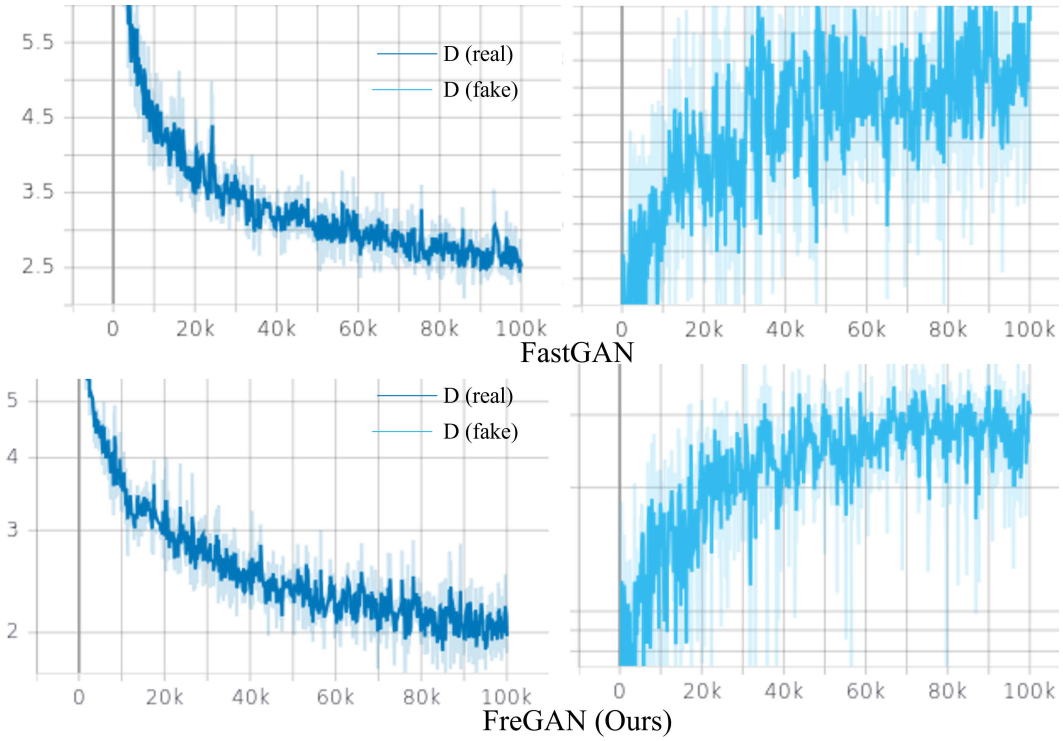


Figure 17: **Comparison results of the outputs of the discriminator throughout the training.** The outputs of the discriminator on fake images of our FreGAN rise more stable and continuous, indicating that our generator is trained more effectively, leading to better generation quality.

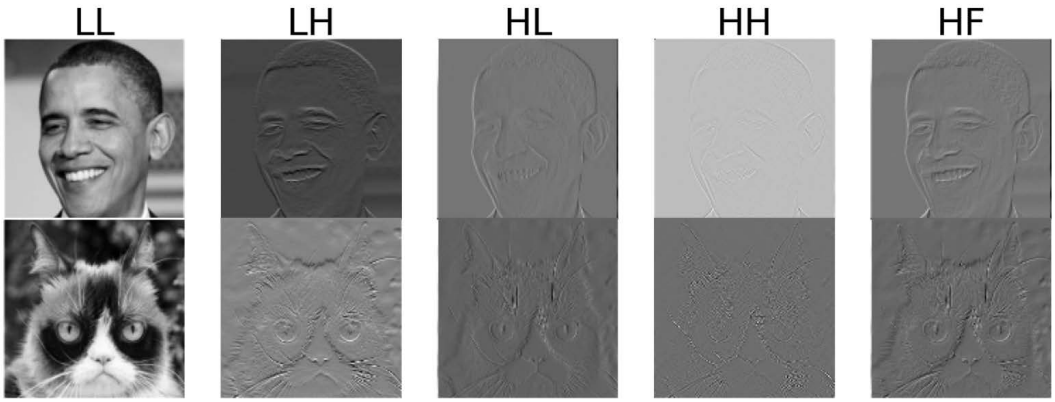


Figure 18: **2D DWT illustration results of differen frequency components.** The low (L) pass filter captures images' overall textures and outlines, and the high (H) pass filter concentrates on details such as the background and edges.

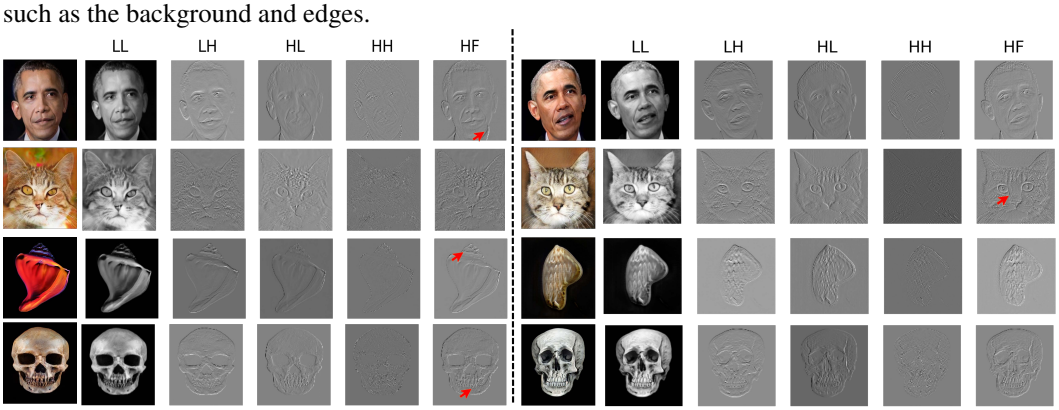


Figure 19: **2D DWT Qualitative comparison results of our FreGAN and the baseline FastGAN.** The images from left to right are generated images, 2D DWT LL, LH, HL, HH, and the combined High-frequency components respectively. Our FreGAN improves the overall quality of generated images and raises the model's frequency awareness, encouraging the generator to produce precise high-frequency signals with fine details.

(Fig. 24); 3) Our FreGAN is stable and can consistently produce effective frequency signals (Fig. 22 and Fig. 23);

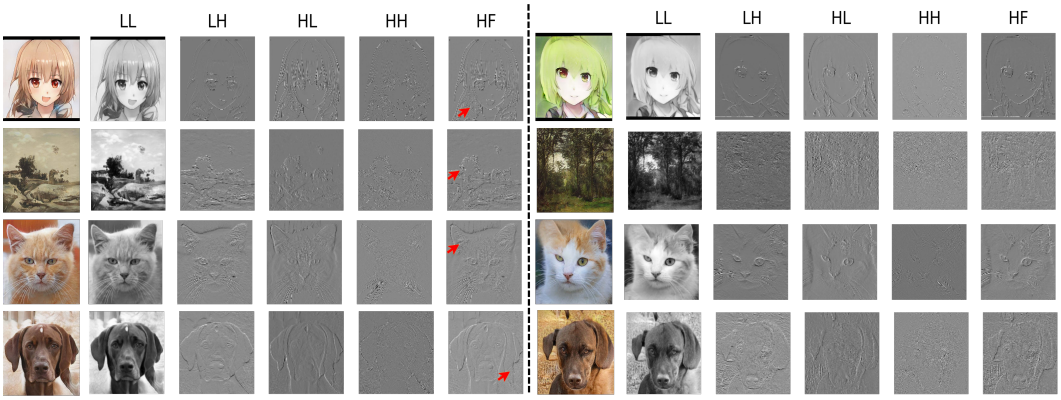


Figure 20: **2D DWT Qualitative comparison results of our FreGAN and the baseline FastGAN.** The images from left to right are generated images, 2D DWT LL, LH, HL, HH, and the combined High-frequency components respectively. Our FreGAN improves the overall quality of generated images and raises the model's frequency awareness, encouraging the generator to produce precise high-frequency signals with fine details.

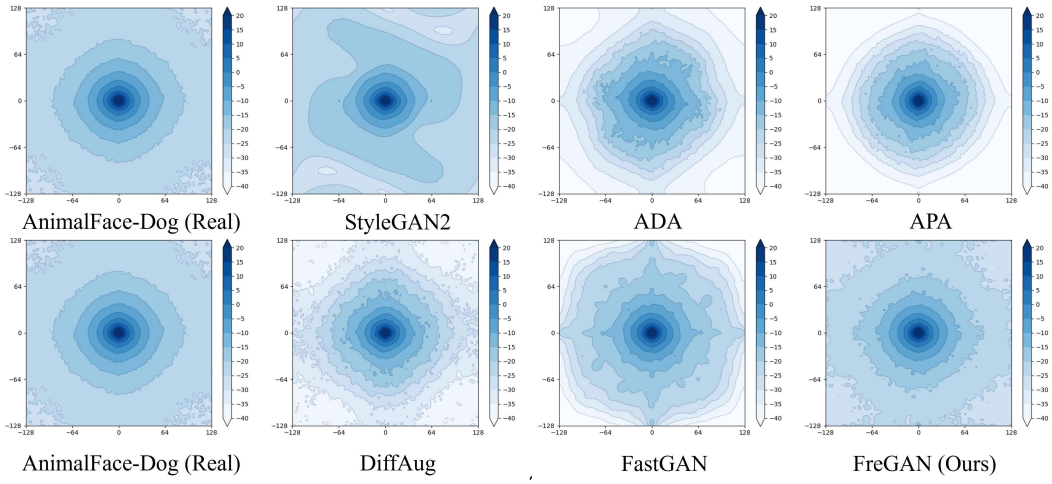


Figure 21: **Comparison results of average 2D power spectrum [18] on Animal Face Dog.** The average 2D power spectrum result for the real data is computed from all training data, and the results of our FreGAN and compared methods are computed from 5k generated images.

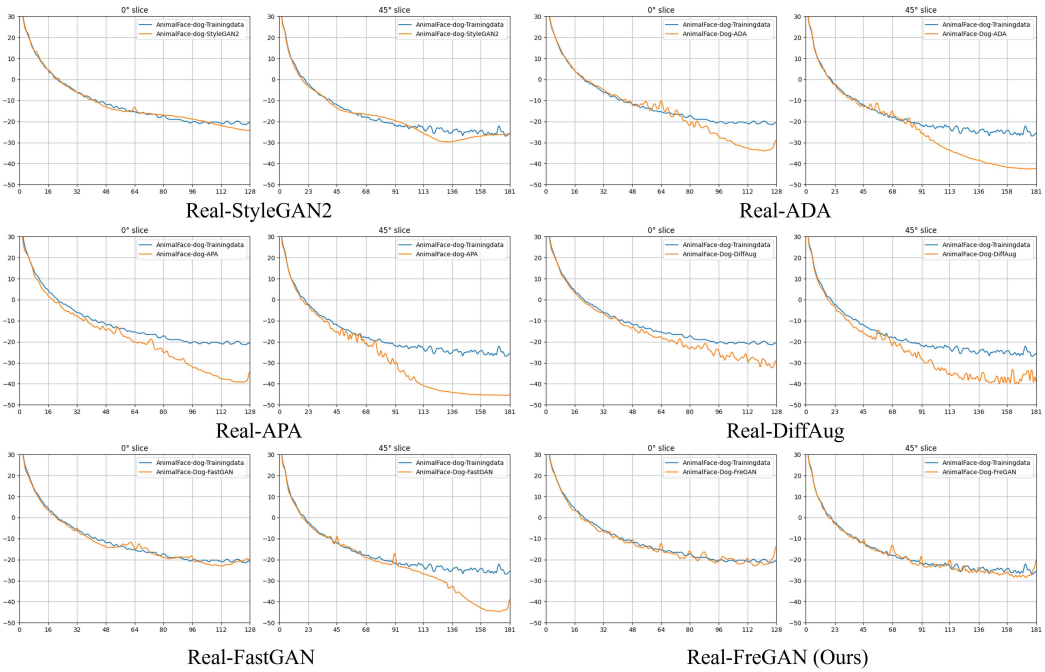


Figure 22: **Comparison results of one-dimensional slices of the power spectrum [18] on Animal Face Dog.** The 1D slices of the spectrum are computed along the horizontal angle (0°) without azimuthal integration [18].

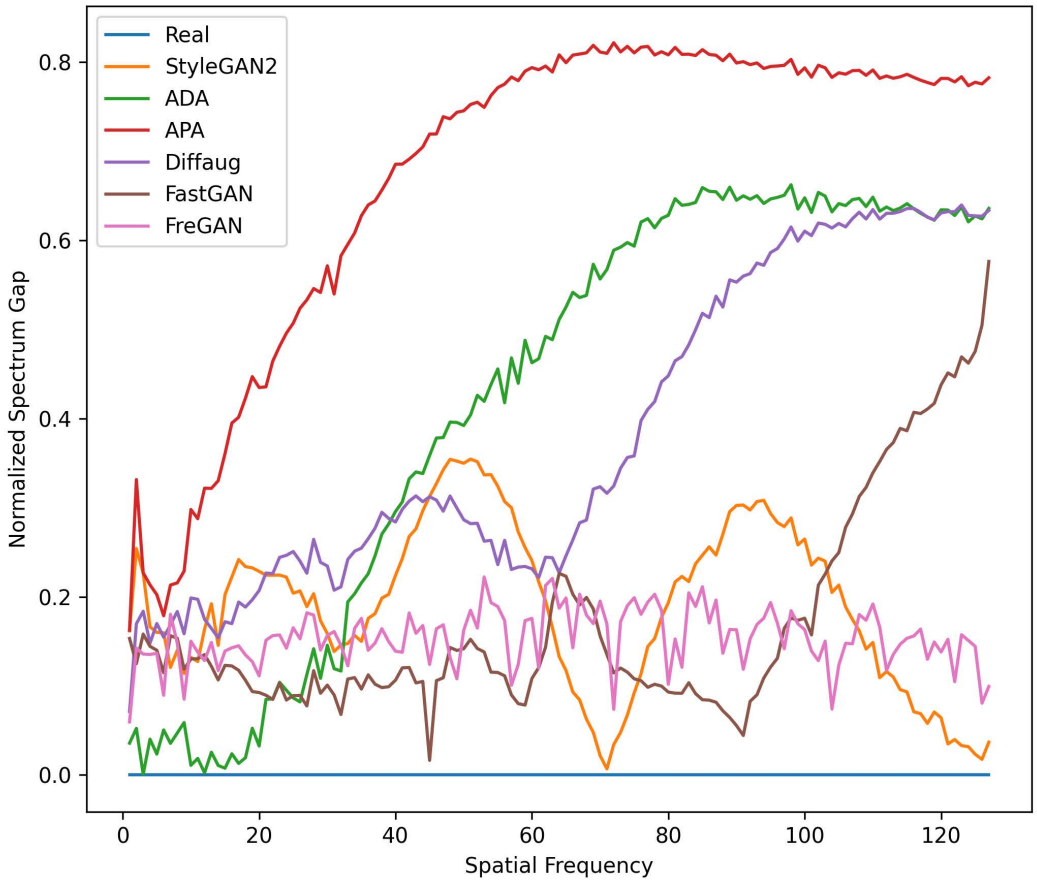


Figure 23: **The power spectrum distance [9] between real images and those generated by our FreGAN and compared methods.** The gap between the images generated by our FreGAN and the real images is the smallest, demonstrating that our FreGAN can produce more realistic frequency signals, especially high-frequency signals as shown in the right half of the figure.

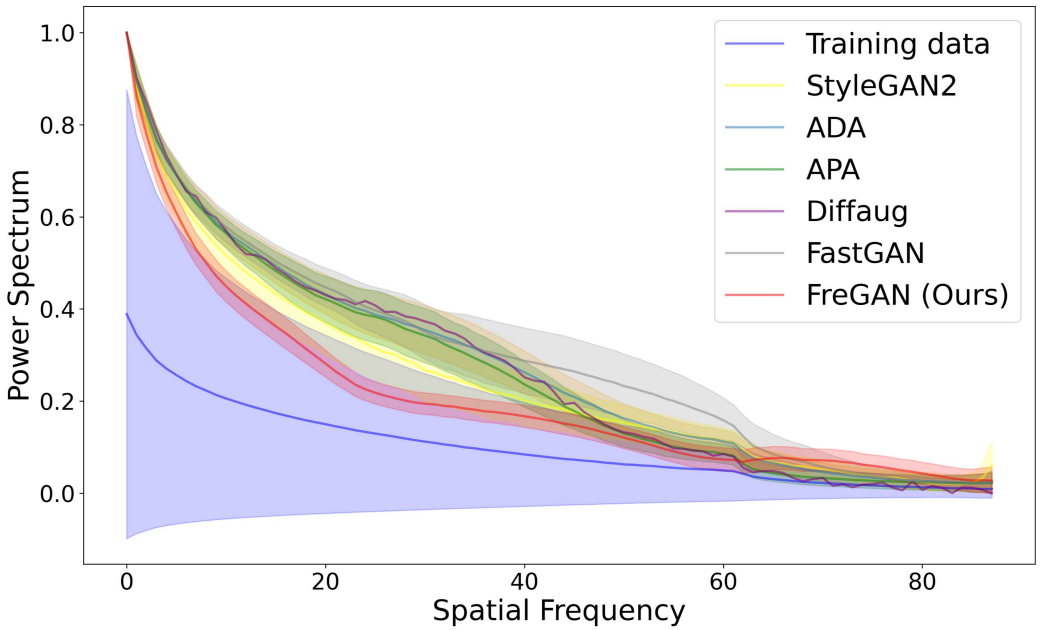


Figure 24: **Power Spectrum [8] comparison results of our FreGAN and baselines on Animal Face Dog dataset.** The statistics (mean and variance) results are obtained after azimuthal integration over the power-spectrum. The statistics of our method overlap the most with that of the training data, indicating that our model is frequency-aware and can produce more realistic frequency signals.