F-Drop&Match: GANs with a Dead Zone in the High-Frequency Domain

Shin'ya Yamaguchi
NTT
shinya.yamaguchi.mw@hco.ntt.co.jp

Sekitoshi Kanai
NTT
sekitoshi.kanai.fu@hco.ntt.co.jp

Abstract

Generative adversarial networks built from deep convolutional neural networks (GANs) lack the ability to exactly replicate the high-frequency components of natural images. To alleviate this issue, we introduce two novel training techniques called frequency dropping (F-Drop) and frequency matching (F-Match). The key idea of F-Drop is to filter out unnecessary high-frequency components from the input images of the discriminators. This simple modification prevents the discriminators from being confused by perturbations of the high-frequency components. In addition, F-Drop makes the GANs focus on fitting in the low-frequency domain, in which there are the dominant components of natural images. F-Match minimizes the difference between real and fake images in the frequency domain for generating more realistic images. F-Match is implemented as a regularization term in the objective functions of the generators; it penalizes the batch mean error in the frequency domain. F-Match helps the generators to fit in the high-frequency domain filtered out by F-Drop to the real image. We experimentally demonstrate that the combination of F-Drop and F-Match improves the generative performance of GANs in both the frequency and spatial domain on multiple image benchmarks (CIFAR, TinyImageNet, STL-10, CelebA, and ImageNet).

1. Introduction

Generative adversarial networks built from deep convolutional networks (GANs) [9, 10, 18, 20] have attracted much attention in the computer vision community and have been utilized in various applications because they can synthesize diverse images with high-fidelity to the target datasets. The training of GANs is formulated as a competitive game played by two neural networks called a generator and a discriminator; the generator is optimized to produce fake images that can fool the discriminator, and the discriminator is optimized to distinguish the real images from the fake images through min-max optimization. In theory, the model replicates training data as the optimal result. However, recent studies have revealed that GANs fail to replicate data in the frequency domain [6, 7]. Durall *et al*. [6] and Frank *et al*. [7] have reported that the frequency characteristics of the generated images in the high-frequency domain are different from those of real images (we refer to this difference as the frequency gap). They have also shown that the generated images can be easily detected as fakes with almost 100% accuracy by assessing the frequency gap. While the previous studies mainly focus on the aliasing caused by upsampling in CNNs as the cause of the frequency gap, modifying the upsampling is insufficient for correcting the flaws in the frequency domain [7]. In this study, we explore another cause of the frequency gaps to reduce them. Since spatial and frequency domains are dual, reducing the frequency gaps can improve the generative performances of GANs in the spatial domain.

We hypothesize that the frequency gap is caused by the sensitivity of the discriminators to the perturbations in the high-frequency domain. In GANs for image generation,

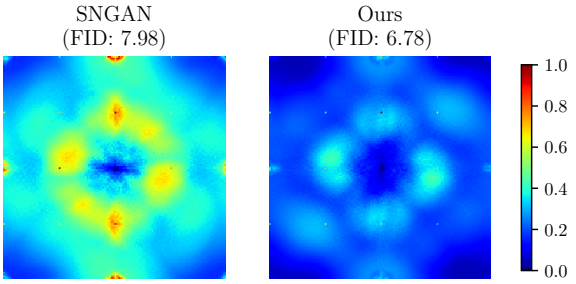

Figure 1. Sensitivities of discriminators in the frequency domain. The sensitivity is measured by single Fourier attack (SFA) [24], which perturbs each frequency component of an image. As the sensitivity, we plot the average differences between outputs of normal and attacked discriminators over 512 images in the CelebA dataset on each pixel. The differences in the low-frequency domain are located near the center of each figure, and the differences in the high-frequency domains are at the edges. Our method outperforms the baseline (SNGAN) in terms of the robustness against the SFA on the high-frequency components and the correctness of the prediction for the low-frequency components.

discriminators are usually implemented as CNN-based binary classifiers. As shown in [24, 28], CNN-based classifiers are sensitive to perturbations of the frequency components. Moreover, Wang *et al*. [25] have reported that CNN-based classifiers predict labels depending on high-frequency components that are hardly recognizable to humans. Accordingly, we conjecture that the discriminators of GANs are also sensitive to the high-frequency components of the input images. Indeed, our experiments demonstrate the sensitivity of the discriminator in the frequency domain: the output of the discriminator is significantly changed by single Fourier attack [24], which perturbs each frequency component of an image (Fig. 1, left). The sensitivity of the discriminators prevents the generators from learning data because the generators are optimized to fool the discriminators by perturbing high-frequency components rather than by replicating data.

To alleviate the sensitivity of the discriminators and the frequency gap, we present two novel techniques, called *frequency dropping* (F-Drop) and *frequency matching* (F-Match). The main idea of F-Drop is to filter out high-frequency components from the inputs of the discriminators (for both real and generated images) and thereby the discriminators concentrate on lower frequency components, which are the dominant components in natural images [27]. We insert a low-pass filter, which filters out frequency components above a certain threshold from images, before the input layer of the discriminators. F-Drop, (i) transforms RGB images into the frequency domain by using discrete cosine transform (DCT), (ii) performs filtering in the frequency domain by element-wise multiplication, and (iii) transforms the images back into RGB space by using inverse discrete cosine transform (IDCT). Since RGB images are used as input, F-Drop does not require any modifications to the original network architectures. By applying F-Drop, the discriminators become robust against high-frequency perturbations (Fig. 1, right), and thus, the generators can dedicate themselves to fooling the discriminators by learning the remaining lower frequency components. However, since F-Drop simply transforms the input of the discriminators, the generators are still free to synthesize the high-frequency components filtered out during the training. Hence, to synthesize realistic frequency components, we propose F-Match, which minimizes the mean error in the frequency domain. F-Match is a simple mini-batch-based regularization term for the objective function of the generators; it can utilize arbitrary frequency transformations (*e.g*., DFT and DCT) and loss functions (*e.g*., the squared and absolute error). We experimentally found that the best function for F-Match is the mean squared error in DCT space. Our experiments show that, in various settings, the combination of F-Drop and F-Match succeeds in synthesizing more realistic images in both the frequency and spatial domains compared with the conventional techniques [4, 6, 7]. Our contributions are summarized as follows:

- We demonstrate that the discriminators of GANs are sensitive to perturbations of high-frequency components through the experiments applying single Fourier attack to discriminators.
- We propose two simple techniques for GANs called F-Drop and F-Match for reducing the frequency gap between real and generated images. F-Drop filters out the high-frequency components from the input images of the discriminators, and F-Match minimizes the mean error in the frequency domain by adding a regularization term to the objective function of the generators.
- We confirm that our methods can improve the quality of the generated images on various image datasets (CIFAR-10/-100, TinyImageNet, STL-10, CelebA, and ImageNet).

2. Related Work

2.1. Frequency Gaps in Generative Models

Frequency gaps in GANs or CNN-based generative models have been studied in recent papers [6, 7]. Durall *et al*. [6] and Frank *et al*. [7] have found that there are frequency gaps between real images and images generated from CNN-based models by using discrete Fourier transform (DFT) and discrete cosine transform (DCT). They have also found that the generated images are detected as fake by linear classifiers trained on the frequency components of the images. These studies have hypothesized that upsampling in CNNs is a cause of the frequency gaps. In particular, Frank *et al*. [7] have shown that the frequency gaps can be reduced by modifying the upsampling in the generators (by using, *e.g*., binomial upsampling). However, the modification of upsampling is not sufficient for generating undetectable fake images in the frequency domain. Furthermore, we empirically report that the modification may degrade the generative performance of GANs in the spatial domain (Sec. 6.4); this degradation has not been discussed in any previous works. In contrast to these works, we show that the discriminators of GANs are sensitive to high-frequency perturbations, and that this sensitivity is also one of the causes of the frequency gaps.

For alleviating frequency gaps, Durall *et al*. [6] have proposed spectral regularization which minimizes the binary cross-entropy between the azimuthal integrals of the real and generated images in the frequency domain. Although spectral regularization has a similar form to F-Match, it minimizes the gaps between each generated image and the mean value of the real images whereas F-Match minimizes the gaps between the mean values of the generated and real

images over each mini-batch. Chen *et al*. [4] have proposed a similar technique called SSD, which modifies discriminators by adding a classifier in the frequency domain and utilizes the output of the classifier to modulate the losses of the GANs. SSD does not use the gradients of the frequency classifier for training of GANs, whereas F-Match directly uses the gradients of the loss in the frequency domain.

2.2. Sensitivity of CNNs for Frequency Components

In the context of adversarial attacks for CNN models, Tsuzuku and Sato [24] have pointed out a sensitivity of CNNs in the frequency domain by conducting an analysis using their own black-box attack called single Fourier attack (SFA). SFA perturbs an image in the directions of each Fourier basis. Similar to [24], Yin *et al*. [28] have shown that naturally trained CNNs are sensitive to high-frequency perturbations. In addition, Wang *et al*. [25] have indicated that the output of CNN-based classifiers depends on the high-frequency components that are not visible to humans. However, they have also shown that dropping the high-frequency components from the training does not degrade the final test performances. Moreover, Xu *et al*. [27] have shown that dropping the high-frequency components from the input images of CNNs by thresholds helps to reduce the input size and improves performance. In summary, the previous results provide two key insights: (i) CNNs-based classifiers have flaws in processing high-frequency components in input images, and (ii) high-frequency components are not essential for training the classifiers. These insights underlie the idea of F-Drop described in Sec. 5.1.

3. Background

3.1. Generative Adversarial Networks

A generative adversarial network is composed of a *generator* network G_θ parametrized by θ, and a *discriminator* network D_ϕ parameterized by ϕ [9]. The G_θ generates a fake sample $x_{\text{fake}} = G_\theta(z)$ from a random noise $z \sim p_z$, and the D_ϕ distinguishes an observation x whether x comes from the data distribution p_{data} or not. The objective functions for training the discriminator and generator are

$$\begin{aligned}\mathcal{L}_{D_\phi} &= -\mathbb{E}_{x\sim p_{\text{data}}} \log D_\phi(x) \\ &\quad -\mathbb{E}_{z\sim p_z} \log\left(1 - D_\phi(G_\theta(z))\right), \qquad (1)\\ \mathcal{L}_{G_\theta} &= -\mathbb{E}_{z\sim p_z} \log D_\phi(G_\theta(z)). \qquad (2)\end{aligned}$$

By training of G_θ and D_ϕ, D_ϕ learns to maximize the probability of assigning a "real" label to real examples and a "fake" label to fake examples, whereas G_θ learns to maximize the probability of D_ϕ's failure of distinction. In theory, when G_θ and D_ϕ converge to the optimal point, the generator network G_θ implicitly replicates p_{data}.

In this paper, we mainly focus on GANs built from CNNs. There are several variants, such as DCGAN [20], WGAN-GP [10], and SNGAN [18]. We can apply F-Drop and F-Match to any of these variants because they are designed as an additional masking layer in discriminators or as an additional regularization term.

3.2. Frequency Transformations

Here, we briefly summarize the foundations of discrete cosine transform (DCT) that is used in F-Drop and F-Match. Note that, for simplicity, our discussion regards transformations of a gray-scale square image $X \in \mathbb{R}^{H\times H}$ but it can be easily extended to color images by performing the same computations on each channel.

Two-dimensional DCT [1, 8] is formulated as follows:

$$C(u,v) = \frac{2\alpha(u)\alpha(v)}{H} \sum_{i=0}^{H-1}\sum_{j=0}^{H-1} X(i,j)c(i,j,u,v), \qquad (3)$$

where $\alpha(0) = 1/\sqrt{2}, \alpha(t) = 1$ (for $t \neq 0$), and

$$c(i,j,u,v) = \cos\left[\frac{(2i+1)u\pi}{2H}\right]\cos\left[\frac{(2j+1)v\pi}{2H}\right].$$

where (i,j) represents a spatial pixel coordinate, (u,v) is a frequency coordinate. This form is called DCT-II. We choose DCT for F-Drop and F-Match as the default because it does not have discontinuous boundaries that produces high-frequency noise, in contrast to DFT [23]. As the transformation from the frequency domain back to the spatial domain, we use two-dimensional inverse discrete cosine transform (IDCT):

$$X(i,j) = \frac{2}{H} \sum_{u=0}^{H-1}\sum_{v=0}^{H-1} \alpha(u)\alpha(v)C(u,v)c(i,j,u,v), \qquad (4)$$

where $\alpha(\cdot)$ and $c(\cdot)$ as the same as in Eq. (3).

4. Analyzing GANs with Single Fourier Attack

We hypothesize that the frequency gap of GANs is caused by the sensitivity of the discriminators to perturbations in the high-frequency domain. To confirm this hypothesis, we analyze GANs subjected to single Fourier attack (SFA) [24]. SFA attacks classification models by perturbing the input images in the Fourier basis directions. For each perturbation, SFA selects a single frequency component and creates striped noise according to the selected component. A perturbation $\delta(u,v)$ of the frequency coordinate (u,v) for an $H \times H$ image is defined as follows:

$$\begin{aligned}\delta(u,v) =& \epsilon((1+\mathrm{j})(F_H)_u \otimes (F_H)_v \\ &+ (1-\mathrm{j})(F_H)_{H-u} \otimes (F_H)_{H-v}),\end{aligned} \qquad (5)$$

where ϵ is a hyperparameter determining the size of the perturbation, F_H is the matrix of the Fourier basis and $(F_H)_i$

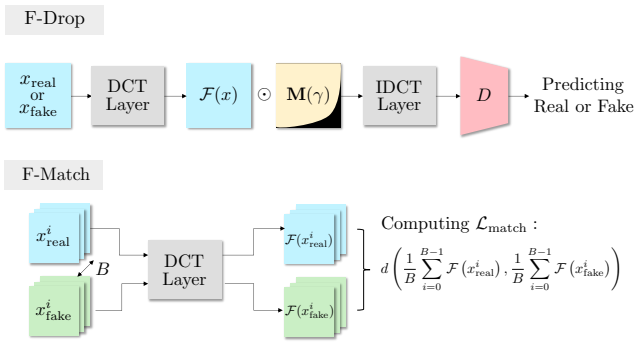

Figure 2. Illustration of proposed methods

represents the i-th row of F_H. Note that $\otimes$ means the Kronecker product and j is the imaginary unit. We will use SFA to investigate the sensitivity of discriminators in the frequency domain.

As a preliminary experiment, we tested ResNet-based SNGANs [18] trained on the CelebA dataset [17] (The details of the training are shown in Sec. 6.1). For SFA, we set ϵ to $10/255$. The left-hand side of Figure 1 visualizes the results of SFA. The visualization procedure followed [24]. Each pixel coordinate corresponds to a frequency component used for SFA, and each pixel represents the absolute differences $|D(x) - D(x + \delta(u, v))|$, *i.e.*, the sensitivity to the perturbation. Note that the differences are normalized to $[0, 1]$ by dividing with the maximum value of the SNGANs and ours. We can see that the SNGANs are sensitive to high frequency perturbations like the results in [28]. This indicates that the discriminators are easily fooled by perturbing the high-frequency domain and their sensitivity in this regard leads to the frequency gaps because the generators focus on synthesizing the high-frequency perturbations rather than a realistic image. Since the high-frequency components are not necessary for training CNNs, as discussed in Sec. 2.2, we will examine a method of filtering out them from the input images.

5. Proposed Method

Figure 2 illustrates the overview of F-Drop and F-Match. F-Drop filters out the high-frequency components from the input images for discriminators, while F-Match is a regularization method for generators, which penalizes the mini-batch mean error in the frequency domain between the real and generated images. F-Drop and F-Match are independent of each other and can be easily incorporated in the architectures of GANs.

5.1. Frequency Dropping

First, we introduce the idea of frequency dropping (F-Drop). As discussed in Sec. 2.2 and 4, the discriminators of GANs are sensitive to the high-frequency components of the input images, but can be trained without the high-frequency components. F-Drop is based on these insights; it filters out the high-frequency components from images by masking with a user-defined threshold parameter $\gamma \in [0, 1]$. The procedure of F-Drop is quite simple: (i) transform an input image into the frequency domain, (ii) drop the high-frequency components, and (iii) transform the frequency components back into the spatial domain. F-Drop transforms an input color image $\mathbb{R}^{3\times H\times W}$ for the discriminators as follows:

$$\text{Drop}(x, \gamma) = \mathcal{F}^{-1}\left(\mathcal{F}(x) \odot \mathbf{M}(\gamma)\right), \tag{6}$$

where $\mathcal{F}$ is a frequency transform function, such as DCT, $\mathcal{F}^{-1}$ is an inverse frequency transform function, such as IDCT, and $\mathbf{M}(\gamma) \in \mathbb{R}^{3\times H\times W}$ is a mask matrix for filtering specific frequency components. Note that $\odot$ denotes element-wise multiplication. We chose DCT to be $\mathcal{F}$ and IDCT to be $\mathcal{F}^{-1}$. An element in a coordinate (c, u, v) of the mask matrix $\mathbf{M}(\gamma)$ is defined as

$$M^c_{u,v}(\gamma) = \begin{cases} 1 & (\sqrt{u^2 + v^2} \le \gamma\sqrt{H^2 + W^2}) \\ 0 & (\sqrt{u^2 + v^2} > \gamma\sqrt{H^2 + W^2}). \end{cases} \tag{7}$$

That is, we drop the high-frequency components of coordinates farther away than $\gamma\sqrt{H^2 + W^2}$, which is the Euclidean distance from the origin point $(0, 0)$ (*i.e.*, the direct current component). After masking, we utilize the remaining lower frequency components for training the GANs (Fig. 2, top). We can adjust the cutoff frequency with the threshold hyperparameter γ. As defined in Eq. (7), the channels c share the mask element, and thus, the $\text{Drop}(\cdot)$ calculation can be implemented by broadcasting a single channel mask $\mathbf{M}(\gamma) \in \mathbb{R}^{H\times W}$. Since all of the operations in $\text{Drop}(\cdot)$ are differentiable with respect to the input data, we can train the models by gradient descent via backpropagation in an end-to-end fashion.

5.2. Frequency Matching

Frequency matching (F-Match) is for minimizing the frequency gap between the real and generated images. The key idea is matching the frequency characteristics of the real and generated images. F-Match minimizes the frequency gaps by using the mini-batch statistics of the images because an image generated from GANs does not have a one-to-one correspondence to a real image. This regularization strategy is commonly used by methods such as feature matching in [22] and MMD-GAN [16]. The loss function of F-Match is formalized as follows:

$$\mathcal{L}_{\text{match}} = d\left(\bar{X}_{\text{real}}, \bar{X}_{\text{fake}}\right), \tag{8}$$

$$\bar{X}_{\text{real}} = \frac{1}{B}\sum_{i=0}^{B-1}\mathcal{F}(x^i_{\text{real}}),\ \bar{X}_{\text{fake}} = \frac{1}{B}\sum_{i=0}^{B-1}\mathcal{F}(x^i_{\text{fake}}),$$

where $d(\cdot)$ is an error function, B is batch size for each training iteration, and x^i_{real} is the i-th real image, and x^i_{fake} is the i-th generated image from the GANs. $d(\cdot)$ and $\mathcal{F}(\cdot)$ can be set to an arbitrary error function (*e.g.*, squared error) and arbitrary frequency transform (*e.g.*, DCT). In the supplementary materials, we evaluate various combinations of $d(\cdot)$ and $\mathcal{F}(\cdot)$ and show that the mean squared error (MSE) in DCT space is the best choice. We use the following MSE-based function:

$$d_{\text{MSE}} = \frac{1}{HW}\sum_{u}^{H}\sum_{v}^{W}\left(\bar{X}_{\text{real}}(u,v) - \bar{X}_{\text{fake}}(u,v)\right)^2, \quad (9)$$

where $\bar{X}(u,v)$ is the (u,v) coordinate of $\bar{X}$. In the optimization, $\mathcal{L}_{\text{match}}$ is added as a regularization term to the objective function defined in Eq. (2):

$$\mathcal{L}_{G_\theta} = \mathbb{E}_{z\sim p_z}\log D_\phi(G_\theta(z)) + \lambda\mathcal{L}_{\text{match}}, \quad (10)$$

where λ is a balancing hyperparameter. As discussed in Sec. 2.1, spectral regularization (SR) [6] is defined in a similar form to F-Match. Following [6], the loss function of SR for $H \times H$ square images is defined as

$$\mathcal{L}_{\text{SR}} = \frac{1}{B}\sum_{i=0}^{B-1} d_{\text{SR}}\left(\bar{X}_{\text{real}}, \mathcal{F}(x^i_{\text{fake}})\right), \quad (11)$$

$$d_{\text{SR}} = -\frac{1}{H/2-1}\sum_{r=0}^{H/2-1}\text{BCE}\big(A(\bar{X}_{\text{real}}, r), A(X_{\text{fake}}, r)\big),$$

where $\text{BCE}(\cdot)$ is the binary cross entropy function and $A(X,r)$ is the azimuthal integral $\frac{1}{2\pi}\int_0^{2\pi}|X(r,\theta)|d\theta$, which approximates 2D DFT images into 1D signals with respect to the radial distance r in polar coordinates (r,θ). Note that SR differs from F-Match in that it uses a single generated image for minimizing the frequency gaps.

The final objective functions using F-Drop and F-Match are:

$$\begin{aligned}\mathcal{L}_{D_\phi} = &-\mathbb{E}_{x\sim p_{\text{data}}}\log D_\phi(\text{Drop}(x,\gamma)) \\ &-\mathbb{E}_{z\sim p_z}\log\left(1 - D_\phi(\text{Drop}(G_\theta(z),\gamma))\right),\end{aligned} \quad (12)$$

$$\mathcal{L}_{G_\theta} = -\mathbb{E}_{z\sim p_z}\log D_\phi(\text{Drop}(G_\theta(z),\gamma)) + \lambda\mathcal{L}_{\text{match}}. \quad (13)$$

The overall training procedure with F-Drop and F-Match is summarized in Algorithm 1. Note that, unlike the training of normal GANs, we pre-fetch the input real images $\{x_i\}$ for calculating the loss function of F-Match (Eq. 8) on line 4. `GetSample` and `GenNoise` are functions for fetching batch images and for generating batch noise from a normal distribution.

Algorithm 1 Training of GAN with F-Drop and F-Match

Require: Batchsize B, learning rate η_θ, η_ϕ, number of critics K, hyperparameters γ, λ

1: Randomly initialize parameters θ, ϕ
2: **while** not convergent **do**
3: **for** $k = 1$ to K **do**
4: $\{x^i_{\text{real}}\}_{i=0}^{B-1} \leftarrow$ `GetSample`(B)
5: $\{z^i\}_{i=0}^{B-1} \leftarrow$ `GenNoise`(B)
6: **if** $k = 1$ **then**
7: $\{x^i_{\text{fake}}\}_{i=0}^{B-1} \leftarrow \{G_\theta(z^i)\}_{i=0}^{B-1}$
8: $\mathcal{L}_{\text{match}} \leftarrow d\left(\frac{1}{B}\sum_{i=0}^{B-1}\mathcal{F}(x^i_{\text{real}}), \frac{1}{B}\sum_{i=0}^{B-1}\mathcal{F}(x^i_{\text{fake}})\right)$
9: $\mathcal{L}_{G_\theta} \leftarrow -\sum_i^B \log D_\phi(\text{Drop}(x^i_{\text{fake}},\gamma))) + \lambda\mathcal{L}_{\text{match}}$
10: $\theta \leftarrow \theta - \eta_\theta\nabla_\theta\mathcal{L}_{G_\theta}$
11: **end if**
12: $\mathcal{L}_{D_\phi} \leftarrow -\sum_{i=0}^{B-1}\log D_\phi(\text{Drop}(x^i_{\text{real}},\gamma))$
13: $\quad -\sum_{i=0}^{B-1}\log(1 - D_\phi(\text{Drop}(G_\theta(z^i),\gamma)))$
14: $\phi \leftarrow \phi - \eta_\phi\nabla_\phi\mathcal{L}_{D_\phi}$
15: **end for**
16: **end while**

6. Experiments

We evaluate our proposed methods (F-Drop and F-Match) by comparing them with naive baselines and the existing methods [4, 6, 7]. We evaluate our methods in terms of (i) quantitative metrics for GANs (main evaluations), (ii) sensitivity to the high-frequency components (frequency sensitivity analysis), (iii) fake detection in the frequency domain (fake detection), and (iv) the quality of the generated images. The supplementary materials contain an ablation study on F-Match and an analysis of hyperparameter sensitivity of F-Drop.

6.1. Setup

Datasets We used the six different image datasets: CIFAR-10 and CIFAR-100 (32×32) [13], TinyImageNet (32×32) [26], STL-10 (48×48) [5], CelebA (128×128) [17], and ImageNet (128×128) [21]. These datasets have been used for testing the benchmarks of GANs [3, 10, 15, 18, 29]. We applied center cropping and resizing to the images of TinyImageNet, CelebA, and ImageNet before the training. For training, we normalized images into a range of $[-1, 1]$.

GAN Baselines As a baseline, we chose spectral normalization GAN (SNGAN) with ResNet-backbone architectures [18]. As additional baselines, we tested Binomial [7], which replaces the bilinear upsampling filters in the generators with the low-pass filters based on a binomial distribution, spectral regularization (SR) [6], which minimizes the gaps of the azimuthal integral in DFT space by using Eq. (11) (see Sec. 5.2), and SSD-GAN [4], which adds a frequency classifier in DFT space (using the azimuthal integral) to the discriminator and utilizes the output of the classifier to modulate the loss functions of the GANs. We used the Binomial-5 kernel, following Frank *et al*. [7]. SR was based on one in the author's public code repository and used

Table 1. Mean frequency gaps between real and fake images

	CIFAR-10	CIFAR-100	TinyImageNet	STL-10	CelebA	ImageNet
SNGAN	6.89	7.01	9.83	4.19	4.49	4.83
Binomial [7]	7.85	5.83	9.96	4.30	4.74	4.55
SR [6]	6.12	6.80	9.77	3.98	4.48	5.70
SSD-GAN [4]	6.39	6.80	9.97	4.59	4.47	4.80
F-Drop	5.94	6.36	9.29	3.87	4.60	5.39
F-Match	4.84	4.87	7.36	4.04	4.46	4.52
F-Drop&Match	**3.93**	**4.16**	**6.49**	**3.86**	**4.43**	**4.41**

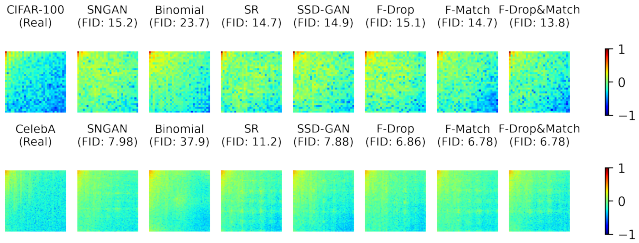

Figure 3. Comparison of average DCT coefficients (top: CIFAR-100, bottom: CelebA). The visualization protocols follow those of Frank *et al*. [7].

λ as 1.0×10^{-5}.[1,2] For SSD-GAN, the implementation was composed of the author's code, and we used $\lambda = 0.5$, following [4].[3] For the labeled datasets (*i.e*., CIFAR-10/-100, TinyImageNet, and ImageNet), we used conditional batch normalization for generators and the projection discriminator following [19]. We evaluate other representative GAN variants, including deep convolutional GAN (DCGAN) [20] and Wasserstein GAN with a gradient penalty (WGAN-GP) [10] instead of SNGAN in Sec. 6.2. We implemented the architectures of GANs with the open-source repository of Lee *et al*. [14].

Training We basically followed the settings of [15]. We trained the GANs for 100k iterations on the datasets except for ImageNet (450k iterations on ImageNet). In all cases, we optimized the GANs with a batch of 64 by using Adam ($\beta_1 = 0, \beta_2 = 0.9$) [12]. The learning rate of the generators and discriminators was 2.0×10^{-4}. As default settings, we selected $\gamma = 0.8$ for F-Drop by searching in $[0.5, 0.9]$. For F-Match, we used $\lambda = 1.0 \times 10^{-2}$ on the 32×32 datasets, $\lambda = 1.0 \times 10^{-4}$ on STL-10 (48×48), $\lambda = 1.0 \times 10^{-5}$ on the 128×128 datasets; we found them by searching in $[1.0 \times 10^{-6}, 1.0 \times 10^{1}]$. The supplementary materials provide details on the hyperparameter search settings. In all experiments, we trained GANs three times, and show the mean and standard deviation of each metric. We evaluated the Fréchet inception distance (FID) after 1k iterations and picked the best FID model. Note that we did not use $\mathbf{M}(\gamma)$ of F-Drop in the evaluations conducted after training.

[1] https://github.com/cc-hpc-itwm/UpConv/

[2] We used PyTorch to implement the differentiable azimuthal integral because the author's implementation, which uses Numpy, is not differentiable. More detailed discussions appear in the supplementary materials.

[3] https://github.com/cyq373/SSD-GAN

6.2. Main Evaluations

Frequency Gaps First, we evaluate the reduction in the frequency gaps. The following total absolute difference in DCT space was used as a measure of the frequency gap:

$$\frac{1}{HW} \sum_{u}^{H} \sum_{v}^{W} \left| \bar{X}_{\text{real}}(u, v) - \bar{X}_{\text{fake}}(u, v) \right|, \qquad (14)$$

where $\bar{X}(u, v)$ is defined in Eq. (8). We computed the gaps between 10k real and generated images, where the real images were randomly selected from each dataset. Table 1 lists the mean frequency gaps measured by Eq. (14). The F-Drop&Match column represents the performances of SNGAN simultaneously applying F-Drop and F-Match. Visualizations of the frequency characteristics are shown in Fig. 3, where the pixels in the upper left represent lower frequency components and ones in the lower right represent higher frequency components. The figure and table show that F-Drop&Match significantly reduced the frequency gaps in all datasets and replicated more realistic frequency characteristics compared with the other methods. In a few cases, F-Drop by itself did not reduce the gaps. This is because F-Drop allows the generators to synthesize the filtered out high-frequency components, and thus, the generated images contain high-frequency components at random. On the other hand, F-Match by itself reduced the gaps in all cases, since it directly minimizes the frequency characteristics. In Fig. 3, the results of F-Match show frequency gaps in the middle range of the frequency domain more so than the results of F-Drop&Match. This is because the generator of F-Match (by itself) focuses on high-frequency components because of the sensitivity of the discriminators to the high-frequency domain. These results indicate that F-Drop&Match reduces the frequency gaps by complementarily combining filtering and direct minimization. Furthermore, F-Drop&Match outperformed the other frequency-oriented methods, *i.e*., Binomial, SR, and SSD-GAN. The poorer performance of these other methods is probably because they do not take account of the sensitivity of the discriminators to the high-frequency domain. The sensitivity of the other methods is discussed in Sec. 6.3. Here, we discuss other reasons why Binomial, SR, and SSD-GAN are inferior to our methods. In the case of Binomial, the binomial upsampling suppresses the high-frequency components in the generators, but does not explicitly regularizes the models to learn the frequency characteristics. Moreover, we found that Binomial tends to degrade the generative performance in the spatial domain (see the evaluations described below). For SR and SSD-GAN, the performance gains sensitively depend on the dataset. This behavior reflects the 1D approximation with the azimuthal integral, which implicitly assumes that the real frequency characteristics are distributed in a concentric pattern in DFT space.

Table 2. Performance comparison on the 32×32 datasets

	CIFAR-10			CIFAR-100			TinyImageNet		
	FID (↓)	KID$_{\times 10^{-3}}$ (↓)	IS (↑)	FID (↓)	KID$_{\times 10^{-3}}$ (↓)	IS (↑)	FID (↓)	KID$_{\times 10^{-3}}$ (↓)	IS (↑)
SNGAN	$14.3^{\pm 0.73}$	$9.20^{\pm 0.57}$	$8.25^{\pm 0.14}$	$15.2^{\pm 0.25}$	$9.76^{\pm 0.35}$	$8.91^{\pm 0.04}$	$21.8^{\pm 3.01}$	$12.3^{\pm 3.43}$	$6.35^{\pm 0.28}$
Binomial [7]	$35.9^{\pm 0.90}$	$21.9^{\pm 1.89}$	$6.60^{\pm 0.15}$	$23.7^{\pm 0.70}$	$14.3^{\pm 0.47}$	$8.09^{\pm 0.08}$	$53.9^{\pm 4.26}$	$30.8^{\pm 8.91}$	$5.34^{\pm 0.29}$
SR [6]	$12.2^{\pm 0.27}$	$7.73^{\pm 2.73}$	$8.43^{\pm 0.01}$	$14.7^{\pm 0.27}$	$9.56^{\pm 0.49}$	$8.94^{\pm 0.05}$	$23.8^{\pm 2.01}$	$18.9^{\pm 2.00}$	$5.96^{\pm 0.25}$
SSD-GAN [4]	$13.4^{\pm 0.13}$	$8.72^{\pm 0.21}$	$8.32^{\pm 0.11}$	$14.9^{\pm 0.88}$	$9.32^{\pm 0.78}$	$9.01^{\pm 0.31}$	$21.1^{\pm 1.51}$	$12.9^{\pm 1.72}$	$6.50^{\pm 0.23}$
F-Drop	$14.1^{\pm 0.81}$	$9.11^{\pm 0.21}$	$8.31^{\pm 0.18}$	$15.1^{\pm 0.15}$	$9.47^{\pm 0.29}$	$8.93^{\pm 0.05}$	$20.4^{\pm 0.46}$	$11.5^{\pm 1.18}$	$6.49^{\pm 0.08}$
F-Match	$12.8^{\pm 0.53}$	$7.90^{\pm 0.32}$	$8.45^{\pm 0.12}$	$14.7^{\pm 0.66}$	$9.09^{\pm 0.89}$	$9.17^{\pm 0.24}$	$20.9^{\pm 0.24}$	$12.7^{\pm 0.46}$	$6.41^{\pm 0.24}$
F-Drop&Match	$\mathbf{10.7^{\pm 0.92}}$	$\mathbf{7.15^{\pm 0.58}}$	$\mathbf{8.45^{\pm 0.06}}$	$\mathbf{13.8^{\pm 0.34}}$	$\mathbf{8.99^{\pm 0.49}}$	$\mathbf{9.16^{\pm 0.00}}$	$\mathbf{18.9^{\pm 1.08}}$	$\mathbf{10.3^{\pm 0.52}}$	$\mathbf{6.55^{\pm 0.14}}$

Table 3. Performance comparison on larger image datasets

	STL-10 (48×48)			CelebA (128×128)			ImageNet (128×128)		
	FID (↓)	KID$_{\times 10^{-3}}$ (↓)	IS (↑)	FID (↓)	KID$_{\times 10^{-3}}$ (↓)	IS (↑)	FID (↓)	KID$_{\times 10^{-3}}$ (↓)	IS (↑)
SNGAN	$34.7^{\pm 1.26}$	$32.0^{\pm 0.91}$	$8.68^{\pm 0.08}$	$7.98^{\pm 0.13}$	$4.45^{\pm 0.42}$	$3.02^{\pm 0.07}$	$62.5^{\pm 1.16}$	$63.5^{\pm 0.80}$	$14.1^{\pm 0.34}$
Binomial [7]	$34.9^{\pm 0.44}$	$32.4^{\pm 1.01}$	$8.66^{\pm 0.09}$	$37.9^{\pm 6.57}$	$22.3^{\pm 0.95}$	$2.86^{\pm 0.02}$	$76.6^{\pm 6.93}$	$74.1^{\pm 6.15}$	$11.6^{\pm 1.14}$
SR [6]	$38.1^{\pm 0.74}$	$34.9^{\pm 0.87}$	$8.49^{\pm 0.02}$	$11.2^{\pm 0.74}$	$5.67^{\pm 0.87}$	$2.91^{\pm 0.06}$	$64.0^{\pm 1.52}$	$64.9^{\pm 2.35}$	$13.9^{\pm 0.49}$
SSD-GAN [4]	$35.6^{\pm 0.25}$	$32.2^{\pm 0.68}$	$8.77^{\pm 0.03}$	$7.88^{\pm 0.64}$	$4.20^{\pm 0.97}$	$3.05^{\pm 0.07}$	$61.2^{\pm 0.49}$	$61.6^{\pm 1.69}$	$14.3^{\pm 0.08}$
F-Drop	$34.7^{\pm 0.75}$	$31.8^{\pm 1.09}$	$8.75^{\pm 0.07}$	$6.86^{\pm 0.47}$	$3.92^{\pm 0.64}$	$3.09^{\pm 0.06}$	$61.0^{\pm 0.59}$	$60.9^{\pm 1.86}$	$14.2^{\pm 0.21}$
F-Match	$34.0^{\pm 0.72}$	$31.1^{\pm 0.76}$	$8.79^{\pm 0.05}$	$6.78^{\pm 0.16}$	$3.73^{\pm 0.18}$	$3.08^{\pm 0.04}$	$62.0^{\pm 1.33}$	$62.2^{\pm 1.35}$	$14.4^{\pm 0.18}$
F-Drop&Match	$\mathbf{33.8^{\pm 0.66}}$	$\mathbf{30.4^{\pm 0.83}}$	$\mathbf{8.85^{\pm 0.15}}$	$\mathbf{6.78^{\pm 0.11}}$	$\mathbf{3.61^{\pm 0.10}}$	$\mathbf{3.16^{\pm 0.05}}$	$\mathbf{60.4^{\pm 0.71}}$	$\mathbf{60.5^{\pm 0.51}}$	$\mathbf{14.5^{\pm 0.30}}$

Table 4. Performance comparison on GAN variants (CIFAR-100)

	DCGAN [20]			WGAN-GP [10]		
	FID (↓)	KID$_{\times 10^{-3}}$ (↓)	IS (↑)	FID (↓)	KID$_{\times 10^{-3}}$ (↓)	IS (↑)
Baseline	$27.2^{\pm 1.15}$	$16.2^{\pm 1.69}$	$7.16^{\pm 0.26}$	$25.2^{\pm 0.20}$	$21.2^{\pm 0.33}$	$7.72^{\pm 0.03}$
Binomial [7]	$49.8^{\pm 3.78}$	$28.6^{\pm 2.38}$	$6.28^{\pm 0.18}$	$25.1^{\pm 0.48}$	$19.8^{\pm 0.77}$	$7.65^{\pm 0.07}$
SR [6]	$40.8^{\pm 2.28}$	$24.4^{\pm 3.07}$	$6.24^{\pm 0.27}$	$40.9^{\pm 3.68}$	$22.7^{\pm 2.84}$	$6.28^{\pm 0.45}$
SSD-GAN [4]	$34.2^{\pm 1.58}$	$18.6^{\pm 1.62}$	$6.53^{\pm 0.23}$	$45.0^{\pm 1.76}$	$30.1^{\pm 1.96}$	$6.04^{\pm 0.32}$
F-Drop	$25.9^{\pm 0.45}$	$15.8^{\pm 0.27}$	$7.15^{\pm 0.05}$	$23.8^{\pm 0.28}$	$19.3^{\pm 0.52}$	$7.85^{\pm 0.01}$
F-Match	$26.5^{\pm 0.73}$	$16.3^{\pm 1.06}$	$7.23^{\pm 0.17}$	$24.9^{\pm 0.14}$	$20.2^{\pm 0.21}$	$7.59^{\pm 0.11}$
F-Drop&Match	$\mathbf{25.2^{\pm 1.17}}$	$\mathbf{15.4^{\pm 0.44}}$	$\mathbf{7.45^{\pm 0.07}}$	$\mathbf{23.9^{\pm 0.53}}$	$\mathbf{18.9^{\pm 0.79}}$	$\mathbf{7.97^{\pm 0.02}}$

Since SSD-GAN does not use the gradients from the frequency classifier for updating the GANs, its performance gains may be unstable. Meanwhile, F-Match directly minimizes the gaps for each frequency component in an end-to-end fashion and performs stably on the various datasets.

FID/KID/IS Second, we measured the Fréchet inception distance (FID) [11], kernel inception distance (KID) [2], and inception score (IS) [22]. We computed these measures on 100k real and generated images for the 128×128 datasets, and 50k real and generated images for the 32×32 datasets and STL-10. Table 2, 3, and 4 show the scores of FID/KID/IS for each combination of dataset, method, and GAN variant. Note that ↓ means lower is better and ↑ means higher is better. F-Drop by itself and F-Match by itself outperformed the baselines in many cases. More importantly, F-Drop&Match performed the best in all cases. Binomial underperformed the baseline in almost all cases. Similar to the evaluation of the frequency gaps, the performances of SR and SSD-GAN sensitively depend on the datasets, while our methods stably outperform the baselines. These results indicate that our methods can flexibly help GANs to replicate the real images in both the frequency and spatial domain.

6.3. Frequency Sensitivity Analysis

As shown in Sec. 4, the discriminators of GANs are sensitive to the perturbations in the high-frequency domain. We evaluate the sensitivity of our methods by conducting an SFA analysis. Figure 4 compares the baselines and our methods in terms of the results of SFA perturbing each frequency component on CelebA. Figure 12 shows the results of SFA on multiple datasets. We used the same visualization protocol as in Sec. 4. We also tested Binomial, SR, and SSD-GAN on CelebA (Fig. 4); the results on the other datasets appear in the supplementary materials and lead to the same conclusions as given below. F-Drop&Match outperformed all of the baselines. Since the robust frequency domains of F-Drop&Match seem to be the union of the robust frequency domains of F-Drop and ones of F-Match, we see that the robustness of F-Drop&Match comes from combining F-Drop and F-Match complementarily. More importantly, in Fig. 4, we see that Binomial, SR, and SSD-GAN are not robust against the low to middle range of the frequency domain. We consider that this is because these methods, unlike F-Drop, feed the whole input images including their high-frequency components into the discriminators, and thus, the discriminators have trouble focusing on the lower frequency domain. These results suggest that the discriminators of F-Drop&Match can focus on the lower frequency and the generators indirectly adjust their training to learn realistic frequency components by combining F-Drop and F-Match complementarily.

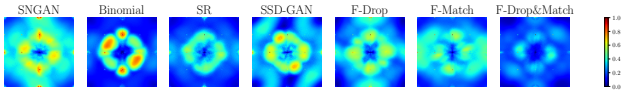

Figure 4. Sensitivity analysis by SFA [24] on CelebA

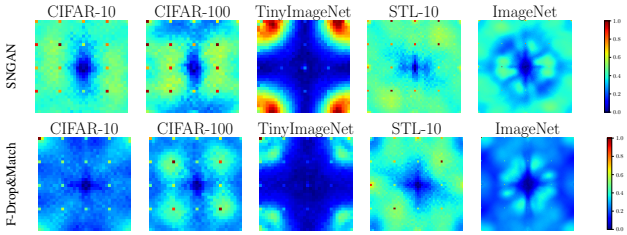

Figure 5. Sensitivity analysis by SFA [24] on multiple datasets

Table 5. Mean accuracy of fake detection with linear binary classification (CIFAR-100)

	Spatial	Frequency
Baseline (SNGAN)	$90.4^{\pm 1.2}$	$92.1^{\pm 1.0}$
Binomial [7]	$95.7^{\pm 0.3}$	$90.9^{\pm 0.5}$
SR [6]	$88.2^{\pm 1.2}$	$91.7^{\pm 0.9}$
SSD-GAN [4]	$89.7^{\pm 1.6}$	$93.2^{\pm 0.6}$
F-Drop	$87.1^{\pm 3.2}$	$89.8^{\pm 2.4}$
F-Match	$81.0^{\pm 1.2}$	$84.7^{\pm 1.3}$
F-Drop&Match	$\mathbf{78.1}^{\pm \mathbf{0.7}}$	$\mathbf{83.1}^{\pm \mathbf{1.9}}$

6.4. Fake Detection

Similar to the evaluation presented in Frank *et al*. [7], we evaluate the detectability of the generated images by using simple linear binary classification models that predict whether an image is real or fake. By measuring the accuracy of these models, we can assess the quality of the generated images in the spatial and frequency domains. The input consisted of pixel values or DCT coefficients of the generated images, and the output was a real value in $[0, 1]$ representing real or fake. Similar to [7], we trained the linear regression model with a batch size of 64 by using Adam ($\beta_1 = 0, \beta_2 = 0.9$, learning rate was 0.001) for 100 epochs. The real images were taken from the CIFAR-100 dataset and the fake images were generated by each method trained on CIFAR-100.

Table 5 lists the mean accuracy of the fake detection models for each setting in CIFAR-100, where Spatial and Frequency represent the results when the pixel values or the DCT coefficients of the generated images are used as the input. Our methods succeeded in degrading the fake detection accuracy in both the spatial and frequency domain; this means they created more realistic images. In addition, the Binomial models slightly degraded accuracy compared with the baseline in the frequency domain but improved accuracy in the spatial domain. This result is consistent with the evaluation in Sec. 6.2: applying a low-pass filter to GAN architectures may lead to difficulty in the training.

Figure 6. Visualization of real and generated images (CelebA)

6.5. Qualitative Results

Lastly, we provide visualizations of the generated images. Figure 6 illustrates the generated images from the SNGAN and F-Drop&Match (ours) models trained on the CelebA dataset. The generated images are randomly selected. We emphasize again that we did not use $\mathbf{M}(\gamma)$ of F-Drop in the evaluation after training. We can see that both SNGAN and ours synthesized rough shapes of human faces that are composed of lower frequency components. Meanwhile, ours was superior to SNGAN at synthesizing more detailed information such as wrinkles and teeth which are composed of higher frequency components, while keeping the information on the lower frequency components such as the positions of facial parts. These results indicate that F-Drop and F-Match make the generators focus on fitting all of frequency components. More importantly, we found that F-Drop produces no visible flaws by filtering the high-frequency components during training. Additional visualization studies including ones on other datasets can be found in the supplementary materials; they show the same tendency as described here.

7. Conclusion

We presented F-Drop and F-Match for minimizing the frequency gaps that appear in images generated by GANs. We demonstrated that the discriminators of GANs are highly sensitive to high-frequency perturbations and the sensitivity can cause frequency gaps. Our methods improve GANs in both the frequency and spatial domain because F-Drop protects the discriminators from high-frequency perturbations and F-Match directly minimizes the frequency gap by using a simple mini-batch error function. Our extensive experiments show that the combination of F-Drop and F-Match outperforms the baselines on various datasets. An important direction of future research will be to introduce adaptive masking without any hyperparameter into F-Drop for filtering effectively and generating realistic images.

Supplementary Materials

This manuscript is the supplementary materials of the main paper (F-Drop&Match: Improved Techniques for GANs in Frequency Domain). We provide (A) the comparative studies of using DCT and DFT for our methods, (B) the ablation studies of F-Match when changing d and $\mathcal{F}$ in Eq. (8) of the main paper, (C) the implementation details of the differentiable azimuthal integral for spectral regularization (SR) [6], (D) the detailed settings and sensitively analysis of the hyperparameter γ and λ, (E) the visual effects caused by F-Drop during training, (F) the additional analysis of the frequency gaps for confirming the validity of the evaluation and the performances of F-Drop in the lower-frequency domain, (G) the additional sensitivity analysis by single Fourier attack [24], (H) the additional visualization studies of the images generated from GANs.

A. Discussion of Frequency Transformations

In this section, we discuss the reason why we use discrete cosine transform (DCT) for F-Drop and F-Match instead of discrete Fourier transform (DFT) that are used in SR [6] and SSD-GAN [4].

Two-dimensional discrete Fourier transform (DFT) for a squared image $X \in \mathbb{R}^{H\times H}$ in the spatial domain is defined as:

$$F(u,v) = \sum_{i=0}^{H-1}\sum_{j=0}^{H-1} X(i,j)\exp\left[-2\pi \mathrm{j}\left(\frac{ui}{H}+\frac{vj}{H}\right)\right], \quad (15)$$

where (i,j) represents a spatial pixel coordinate, (u,v) is a frequency coordinate, and j is an imaginary unit. By Euler's formula ($\exp(\mathrm{j},\theta) = \cos\theta + \mathrm{j}\sin\theta$), DFT represents an input image with complex values composed of periodic (*i.e.*, sine and cosine functions). We can translate that DFT treats an input signal as two-dimensional periodic functions represented by extensively tiling the image in the spatial domain. Thus, DFT produces high-frequency distortions because of the discontinuous boundaries derived from the tiling; this is known as *end effects* of DFT [23]. For avoiding the end effects, we use DCT, which does not have the discontinuous boundaries [23]. In contrast to DFT, DCT represents an input image by only cosine functions of real values. Thus, we can say that DCT treats an input signal as two-dimensional periodic functions represented by symmetrically tiling the image, *i.e.*, DCT does not have the discontinuous boundaries by definition. We experimentally confirm the performance gaps between DFT and DCT in Sec. B.

B. Ablation Study of F-Match

Here, we provide the ablation study for F-Match testing the multiple combinations of the error function $d(\cdot)$ and the frequency transformation $\mathcal{F}(\cdot)$ (*e.g.*, DFT and DCT) in Eq. (8) of the main paper. We basically share the settings of training and network architectures with Section 6 of the main paper.

Table 6. Comparison among F-Match family (CIFAR-100)

	FID (↓)	KID$_{\times 10^{-3}}$ (↓)	IS (↑)
Baseline (SNGAN)	$15.2^{\pm 0.25}$	$9.76^{\pm 0.35}$	$8.91^{\pm 0.04}$
MSE (Pixel)	$15.3^{\pm 0.26}$	$9.67^{\pm 0.29}$	$8.99^{\pm 0.12}$
MSE (DFT)	$15.0^{\pm 0.36}$	$9.20^{\pm 0.24}$	$9.06^{\pm 0.10}$
MSE (DCT)	$\mathbf{14.7^{\pm 0.66}}$	$\mathbf{9.09^{\pm 0.89}}$	$\mathbf{9.17^{\pm 0.24}}$
MAE (DCT)	$14.9^{\pm 0.07}$	$9.40^{\pm 0.84}$	$9.01^{\pm 0.00}$
MKL (DCT)	$15.5^{\pm 0.24}$	$9.89^{\pm 0.05}$	$9.01^{\pm 0.10}$
MSSE (DCT)	$14.8^{\pm 0.23}$	$9.17^{\pm 0.41}$	$9.12^{\pm 0.11}$

As defined in Eq. (8) of the main paper, F-Match can equip arbitrary error function d and frequency transforms $\mathcal{F}$. We explore multiple combinations of d and $\mathcal{F}$ for F-Match. We tested DFT, DCT and Pixel (identity function) as $\mathcal{F}$ and the following four error functions as d: MSE, mean absolute error (MAE), mean KL-divergence (MKL), MSE with concatenating mean and standard deviation of batch frequency components (MSSE). In Table 6, we summarize the ablation study of F-Match. Among the variations, MSE in DCT spaces achieved the best performance in terms of FID/KID/IS. We confirm that minimizing the gap in the frequency domain by using DFT or DCT helps boost the generative performance of GANs whereas minimizing the gap in the spatial domain (Pixel) does not change the performance. In comparison among frequency transforms, DCT is superior to DFT as we expected in Sec. A. Further, in comparison among error functions, we confirm MSE is the best choice.

C. Differentiable Azimuthal Integral

In the main paper, we used the differentiable version of spectral regularization (SR). We reimplemented the differentiable SR with PyTorch because the original reproduction code of azimuthal integral that is published by the author of [6] was implemented by Numpy, *i.e.*, it was not differentiable.[4] For confirming the validity of the reimplementation, we show the reimplementation code of the differentiable azimuthal integral and the comparison results of the non-differentiable and differentiable versions. The reimplementation code was basically constructed by replacing the Numpy functions in the original code with the corresponding PyTorch functions. We tested the performances with SNGAN and CIFAR-100 as well as Section 6 of the main paper. We used the original code of [6] as the non-differentiable version. Algorithm 2 shows the code and Table 7 lists the performance comparison. In Table 7, our differentiable SR succeeded to outperform the baseline

[4] https://github.com/cc-hpc-itwm/UpConv

Algorithm 2 Azimuthal Integral in PyTorch

```
def azimuthal_integral(fft_image, center=None):
    # Calculate the indices from the image
    # These indices are ok to be numpy array
    x, y = np.indices(list(fft_image.shape))
    x, y = torch.from_numpy(x).cuda(), torch.from_numpy(
        y).cuda()

    if not center:
        center = torch.tensor([(x.max() - x.min()) /
            2.0, (y.max() - y.min()) / 2.0])

    r = torch.hypot(x - center[0], y - center[1])

    # Get sorted radii
    ind = torch.argsort(r.flatten())
    r_sorted = r.flatten()[ind]
    i_sorted = fft_image.flatten()[ind]

    # Get the integer part of the radii (bin size = 1)
    r_int = r_sorted.int()

    # Find all pixels that fall within each radial bin.
    deltar = r_int[1:] - r_int[:-1]
    rind = torch.where(deltar)[0]
    nr = rind[1:] - rind[:-1]

    # Cumulative sum to figure out sums for each radius
        bin
    csim = torch.cumsum(i_sorted, dim=0, dtype=torch.
        float32)
    tbin = csim[rind[1:]] - csim[rind[:-1]]

    radial_prof = tbin / nr

    return radial_prof
```

Table 7. Comparison of differentiable and non-differentiable implementation of SR (CIFAR-100)

	FID (↓)	KID$\times 10^{-3}$ (↓)	IS (↑)
Baseline (SNGAN)	$15.2^{\pm 0.25}$	$9.76^{\pm 0.35}$	$8.91^{\pm 0.04}$
Non-Differentiable SR	$15.8^{\pm 0.11}$	$9.81^{\pm 0.54}$	$8.85^{\pm 0.09}$
Differentiable SR (our reimpl.)	$14.7^{\pm 0.27}$	$9.56^{\pm 0.49}$	$8.94^{\pm 0.05}$

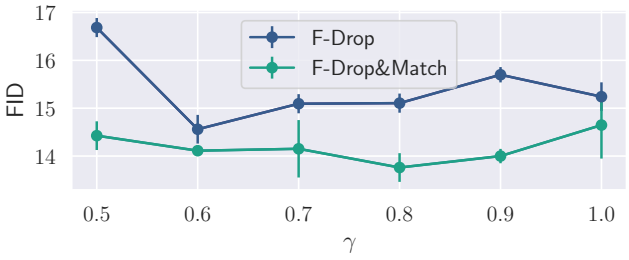

Figure 7. Effect of hyperparameter γ in F-Drop (CIFAR-100)

whereas the non-differentiable SR did not. This result suggests that our reimplementation has a certain validity.

D. Details of Hyperparameter Search

In this section, we describe the details of the hyperparameter search of γ and λ in F-Drop and F-Match. We also show the sensitivity analysis when changing the hyperparameters.

For γ, we searched the values in $\{0.5, 0.6, 0.7, 0.8, 0.9\}$ with SNGAN on CIFAR-100. Fig. 7 illustrates the sensitivity to γ ($\gamma = 1.0$ means the baseline models). In

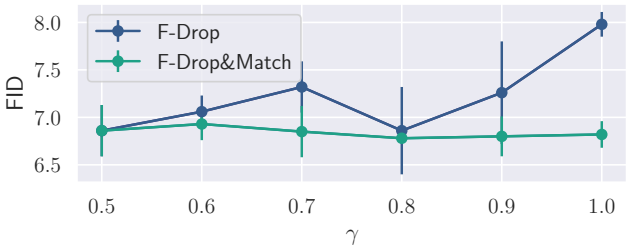

Figure 8. Effect of hyperparameter γ in F-Drop (CelebA)

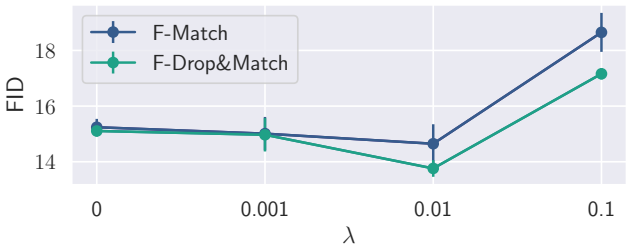

Figure 9. Effect of hyperparameter λ in F-Match (CIFAR-100)

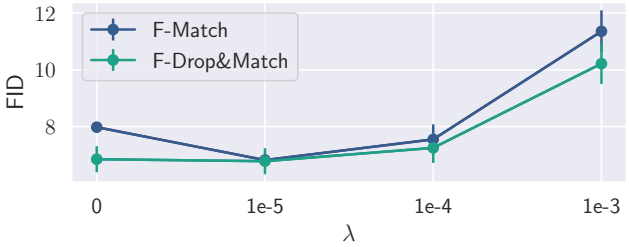

Figure 10. Effect of hyperparameter λ in F-Match (CelebA)

both CIFAR-100 and CelebA, the best γ was 0.8 for F-Drop&Match. The models of F-Drop were inferior to the baselines (SNGANs) in some cases. This is because the generators of F-Drop synthesize filtered out high-frequency components at random, and thus, the high-frequency components may prevent the training. In contrast, the F-Drop&Match models stably outperformed the baselines and F-Drop models with the same γ. Furthermore, we can confirm that there is a difference between single F-Drop and F-Drop&Match in the tendencies; the best γ were 0.5 or 0.6 for F-Drop by itself and 0.8 for F-Drop&Match. This implies that F-Match helps generators to synthesize more realistic high-frequency components that were not learned well by the F-Drop models with $\gamma = 0.8$.

For λ, we searched the values in $\{1.0 \times 10^{-6}, 1.0 \times 10^{-5}, 1.0 \times 10^{-5}, 1.0 \times 10^{-4}, 1.0 \times 10^{-3}, 1.0 \times 10^{-2}, 1.0 \times 10^{-1}, 1.0 \times 10^{0}, 1.0 \times 10^{1}\}$ with SNGAN on each dataset. Fig. 9 and 10 illustrate the sensitivity analysis of λ on CIFAR-100 and CelebA ($\lambda = 0$ means the baseline models). We can see that the relatively small values contributed to improving the baseline in both single F-Match and F-Drop&Match. In contrast to the case of γ, the best values of

Table 8. Frequency gaps among real datasets

	CIFAR-10	CIFAR-100	TinyImageNet	CelebA	ImageNet
CIFAR-10	2.99	3.46	4.40	N/A	N/A
CIFAR-100	3.46	3.03	4.47	N/A	N/A
TinyImageNet	4.40	4.47	3.24	N/A	N/A
CelebA	N/A	N/A	N/A	2.95	3.92
ImageNet	N/A	N/A	N/A	3.92	3.21

Table 9. Frequency gaps in the lower frequency domain

	CIFAR-100		CelebA	
	All-band	Lower-band ($\gamma = 0.8$)	All-band	Lower-band ($\gamma = 0.8$)
SNGAN	7.01	5.06 (-1.95)	4.49	4.06 (-0.43)
Binomial [7]	5.83	4.55 (-1.28)	4.74	4.22 (-0.52)
SR [6]	6.80	4.75 (-2.05)	4.48	4.22 (-0.26)
SSD-GAN [4]	6.80	4.95 (-1.85)	4.47	4.11 (-0.36)
F-Drop	6.36	4.74 (-1.62)	4.60	4.05 (-0.55)
F-Match	4.87	3.97 (-0.90)	4.46	4.04 (-0.42)
F-Drop&Match	**4.16**	**3.80** (-0.36)	**4.43**	**3.98** (-0.45)

λ are different between CIFAR-100 and CelebA. The best values of λ highly depend on the resolution of the input images because the scale of the adversarial losses are changed by the logit size of the discriminators that is different by the resolution. Thus, the best values of λ are transferable across the same resolution datasets (*e.g.* $\lambda = 1.0 \times 10^{-2}$ for 32×32 datasets and $\lambda = 1.0 \times 10^{-5}$ for 128×128).

E. Visual Effects by F-Drop during Training

Here, we discuss the visual effects in the spatial domain of input images by applying F-Drop. Figure 11 illustrates the effects of F-Drop on the spatial domain and frequency domain when changing the threshold parameter γ. In all cases except for $\gamma = 0.0$, F-Drop kept most of the spatial information even it filtered out the higher frequency domain. This indicates that F-Drop does not cause the negative effects during the training of GANs.

F. Detailed Analysis of Frequency Gaps

We provide additional results of the frequency gaps in terms of (i) the validity of the evaluations by the frequency gaps, and (ii) the comparison of the frequency gaps in the lower frequency domain.

First, we confirm the validity of the measurement of the frequency gaps computed by the mean absolute error defined in Eq. (14) of the main paper. To this end, we computed the frequency gaps among the real datasets with respect to the same resolution, *e.g.*, the frequency gaps between CIFAR-10 and TinyImageNet. Table 8 lists the gaps among the real datasets. We used randomly sampled 10,000 images for each dataset by the same protocol in Sec. 6.2 of the main paper. Note that we measured the gaps between the same datasets (*e.g.*, CIFAR-10 and CIFAR-10) by using the two different randomly sampled subsets. The gaps between the real images were in a similar range to the gaps between the real and fake images in Table 1 of the main paper. Furthermore, we see that F-Drop&Match can reduce the gaps at the level of the gaps between real images, *e.g.*, in CIFAR-100, 4.16 of F-Drop&Match is smaller than 4.40 of TinyImageNet. These results indicate that the mean absolute error is reasonable for measuring the frequency gaps and our method can reduce the gaps to be comparable with the gaps between real datasets.

Next, we show the detailed analysis of the frequency gaps in the lower frequency domain. In Table 1 of the main paper, we confirm that the models of F-Drop do not reduce the gaps in some cases (*e.g.*, CelebA). We hypothesize that this is because F-Drop allows the generators to synthesize the filtered out high-frequency components at random. If this hypothesis is true, the gaps should be reduced when they are measured in the lower-frequency domain without the filtered out high-frequency components. Table 9 lists the gaps in the lower-frequency domain. We measured the gap by filtering out the high-frequency components of input images with the mask matrix $\mathbf{M}(\gamma)$ in Eq. (7) of the main paper (denoted as Lower-band ($\gamma = 0.8$)). We used $\gamma = 0.8$ that is the same parameter used in the training of F-Drop by itself and F-Drop&Match. The columns of All-band represent the gaps in all frequency band, and they are reprinted from Table 1 of the main paper. The inside values in the parenthesis of the columns of Low-band are the differences between the Lower-band and All-band values. The gaps of Lower-band were entirely smaller than that of All-band. In particular, the Lower-band gaps of F-Drop by itself were significantly reduced from All-band. Furthermore, we see that F-Drop by itself succeeded in outperforming the baselines in the Lower-band setting. These results suggest that F-Drop makes GANs concentrate on the training of the lower-frequency components.

G. Additional Results of Single Fourier Attack

In Fig. 12, we provide the additional results of single Fourier attack (SFA) except for the results shown in Sec. 6.3 of the main paper. We used the same visualization protocols as Sec. 6.3 of the main paper. In all cases, our F-Drop&Match succeeded to suppress the sensitivity to high-frequency perturbations as well as the main paper. From the results, we consider that combining F-Drop and F-Match is quite important for the discriminators to be robust in the frequency domain.

H. Additional Qualitative Results

We visualize the generated images from SNGAN and our F-Drop&Match for each dataset. Figure 13, 14, 15, 16, and 17 illustrates the images. Note that these images are randomly sampled, not cherry-picked. As we discussed in Sec. 6.5 of the main paper, we can confirm our F-Drop&Match succeed to synthesize detailed (high-frequency) information of images, *e.g.*, human faces and in

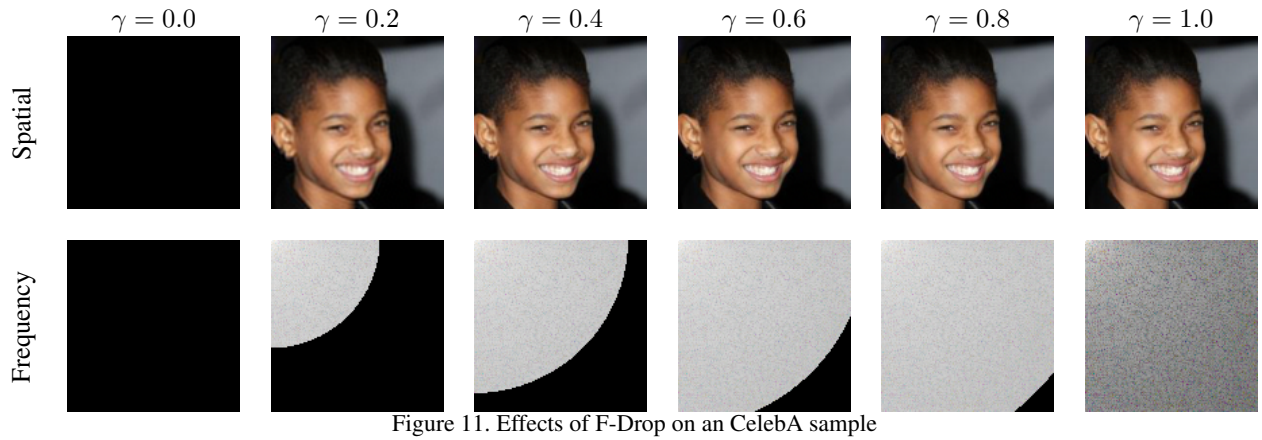

Figure 11. Effects of F-Drop on an CelebA sample

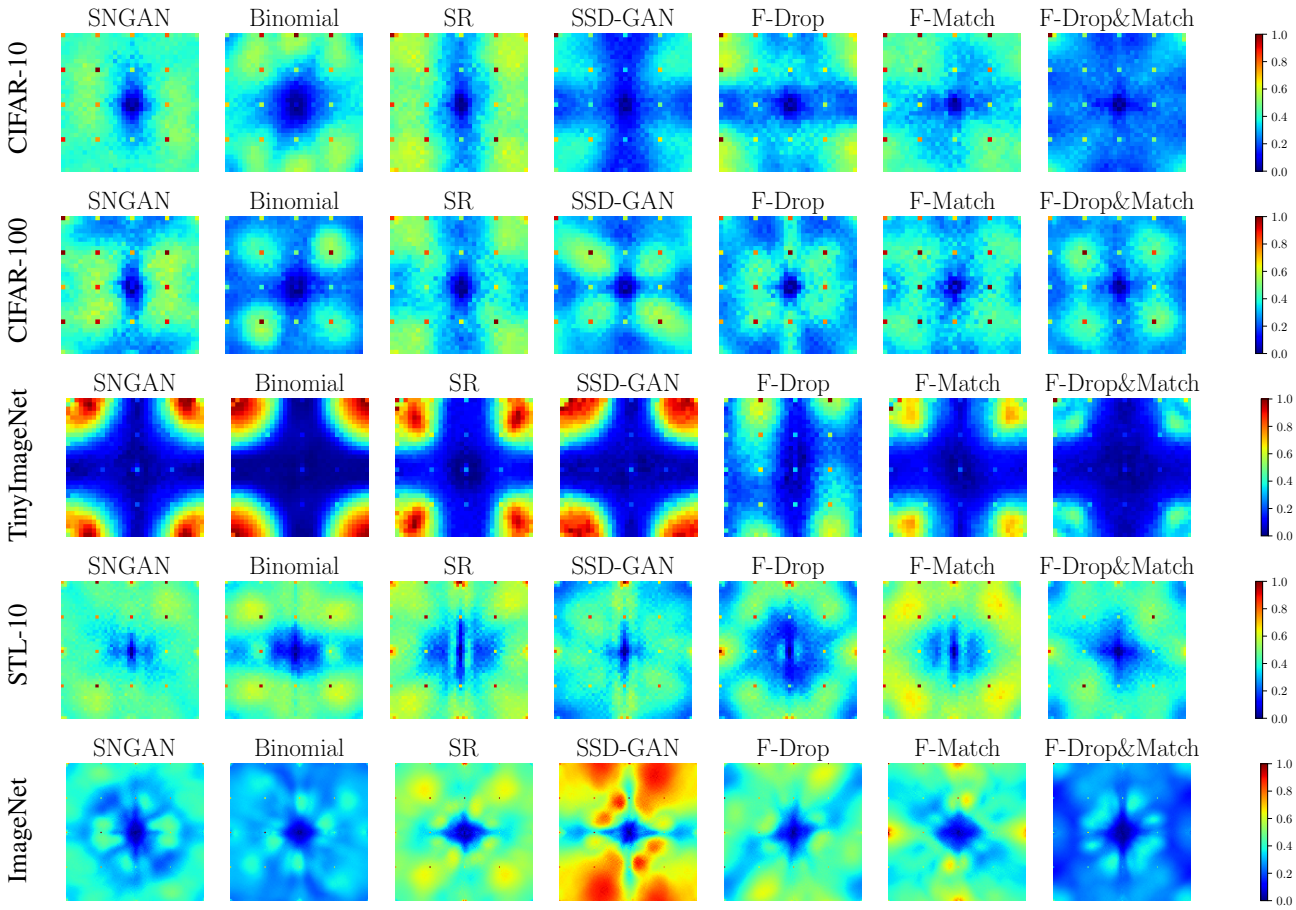

Figure 12. Sensitivity analysis by SFA [24] on multiple datasets

CIFAR-100 and textures of animal skins in STL-10.

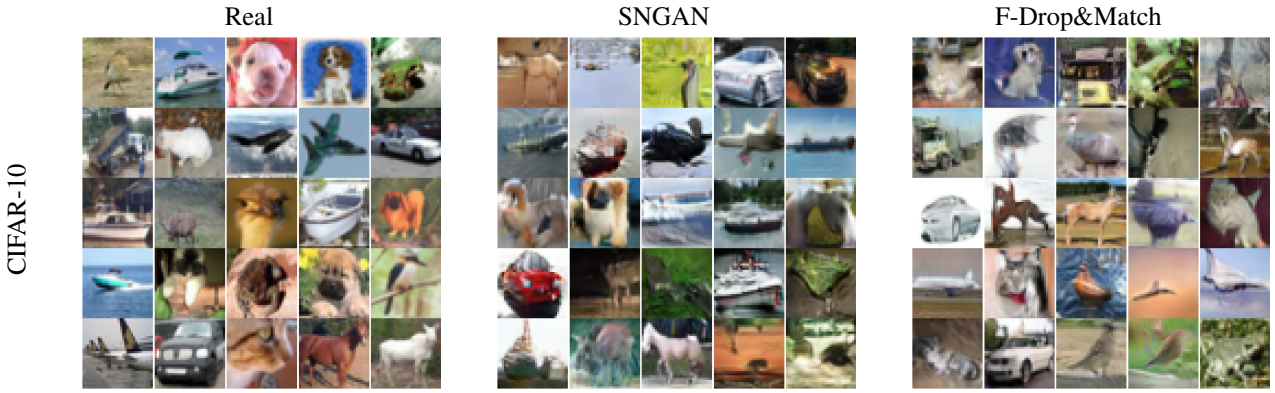

Figure 13. Generated images on CIFAR-10

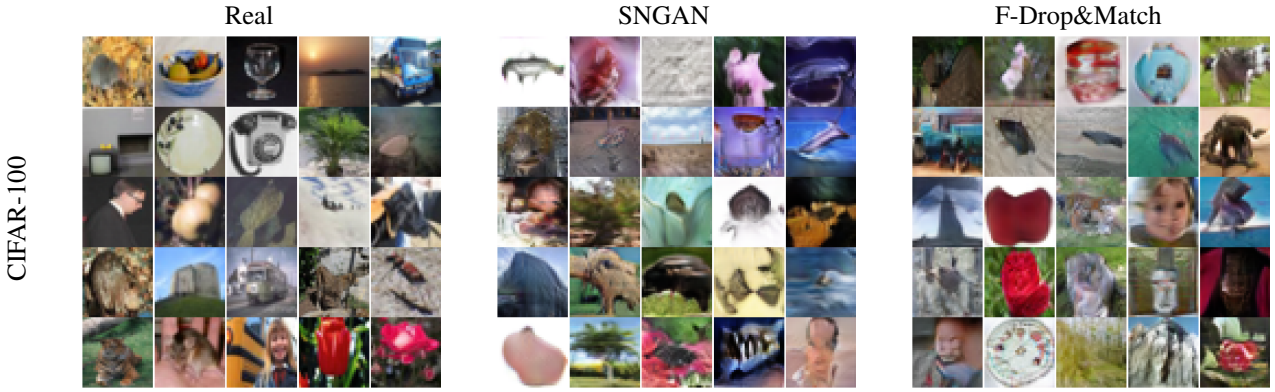

Figure 14. Generated images on CIFAR-100

Figure 15. Generated images on TinyImageNet

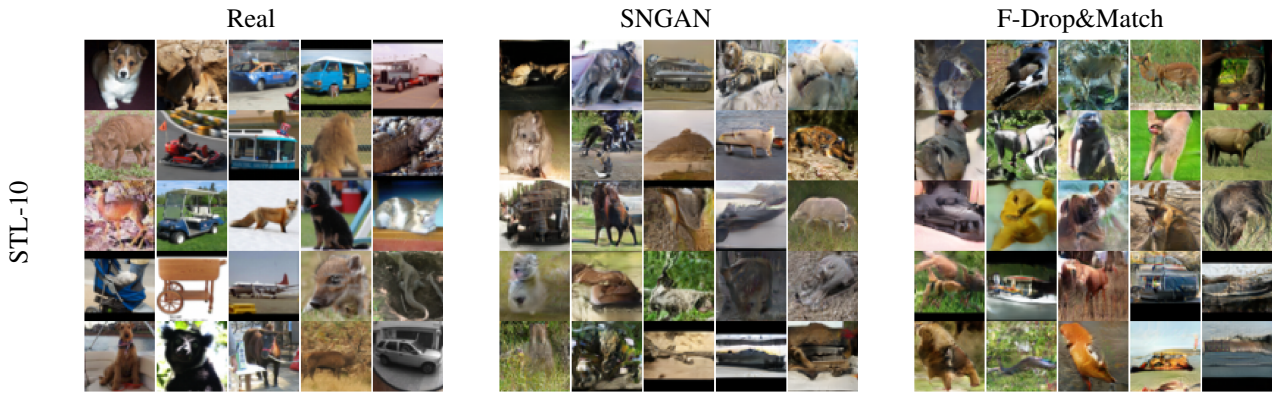

Figure 16. Generated images on STL-10

Figure 17. Generated images on ImageNet

References

[1] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974. 3

[2] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. 7

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 5

[4] Yuanqi Chen, Ge Li, Cece Jin, Shan Liu, and Thomas Li. Ssd-gan: Measuring the realness in the spatial and spectral domains, 2020. 2, 3, 5, 6, 7, 8, 9, 11

[5] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 5

[6] 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*, 2020. 1, 2, 5, 6, 7, 8, 9, 11

[7] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. 2020. 1, 2, 5, 6, 7, 8, 11

[8] Rafael C. Gonzalez and Paul A Wintz. *Digital image processing*. Addison-Wesley Pub. Co., Advanced Book Program Reading, Mass, 1977. 3

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*. 2014. 1, 3

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, 2017. 1, 3, 5, 6, 7

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017. 7

[12] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014. 6

[13] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 5

[14] Kwot Sin Lee and Christopher Town. Mimicry: Towards the reproducibility of gan research. 2020. 6

[15] Kwot Sin Lee, Ngoc-Trung Tran, and Ngai-Man Cheung. Infomax-gan: Improved adversarial image generation via information maximization and contrastive learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3942–3952, January 2021. 5, 6

[16] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabas Poczos. Mmd gan: Towards deeper understanding of moment matching network. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2017. 4

[17] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 4, 5

[18] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *International Conference on Learning Representations*, 2018. 1, 3, 4, 5

[19] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. *International Conference on Learning Representations*, 2018. 6

[20] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. 2015. 1, 3, 6, 7

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 2015. 5

[22] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems 29*. 2016. 4, 7

[23] Jose Tribolet and Ronald Crochiere. Frequency domain coding of speech. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(5):512–530, 1979. 3, 9

[24] Yusuke Tsuzuku and Issei Sato. On the structural sensitivity of deep convolutional networks to the directions of fourier basis functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 51–60, 2019. 1, 2, 3, 4, 8, 9, 12

[25] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8684–8694, 2020. 2, 3

[26] Jiayu Wu, Qixiang Zhang, and Guoxi Xu. Tiny imagenet challenge. Technical report, 2017. 5

[27] Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. Learning in the frequency domain. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 3

[28] Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *Advances in Neural Information Processing Systems*, 2019. 2, 3, 4

[29] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. In *International Conference on Learning Representations*, 2020. 5