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

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 detailed information of training settings, (C) the ablation studies of F-Match when changing d and $\mathcal{F}$ in Eq. (8) of the main paper, (D) the implementation details of the differentiable azimuthal integral for spectral regularization (SR) [2], (E) the detailed settings and sensitively analysis of the hyperparameter γ and λ, (F) the visual effects caused by F-Drop during training, (G) 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, (H) the results of fake image detection evaluation, (I) the additional sensitivity analysis by single Fourier attack [7], (J) 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 [2] and SSD-GAN [1].

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

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

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 [6]. For avoiding the end effects, we use DCT, which does not have the discontinuous boundaries [6]. 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. C.

B. Detailed Training Settings

We basically followed the settings of [5]. 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 ($\beta_1 = 0, \beta_2 = 0.9$) [4]. 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 Frćhet 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.

C. 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.

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 1, we summarize

Table 1. 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)	$14.8^{\pm 0.23}$	$9.17^{\pm 0.41}$	$9.12^{\pm 0.11}$

Table 2. 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.)	$14.7^{\pm 0.27}$	$9.56^{\pm 0.49}$	$8.94^{\pm 0.05}$

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.

D. 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 [2] was implemented by Numpy, *i.e.*, it was not differentiable.[1] 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 [2] as the non-differentiable version. Algorithm 1 shows the code and Table 2 lists the performance comparison. In Table 2, our differentiable SR succeeded to outperform the baseline whereas the non-differentiable SR did not. This result suggests that our reimplementation has a certain validity.

Algorithm 1 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
```

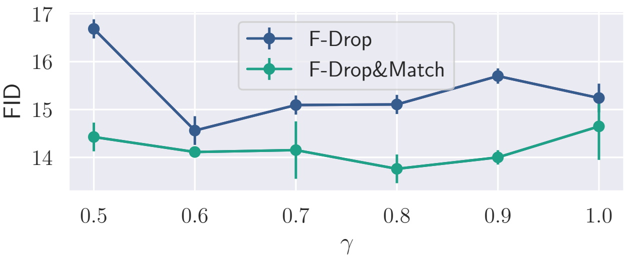

Figure 1. Effect of hyperparameter γ in F-Drop (CIFAR-100)

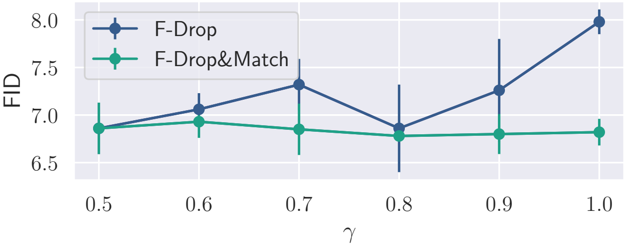

Figure 2. Effect of hyperparameter γ in F-Drop (CelebA)

E. 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 $\{0.5, 0.6, 0.7, 0.8, 0.9\}$

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

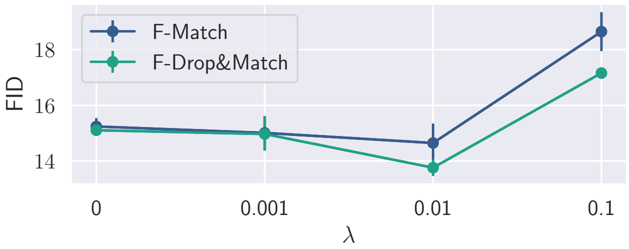

Figure 3. Effect of hyperparameter λ in F-Match (CIFAR-100)

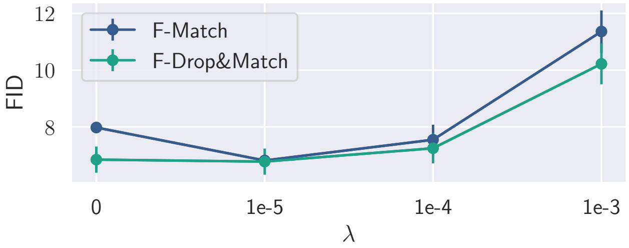

Figure 4. Effect of hyperparameter λ in F-Match (CelebA)

with SNGAN on CIFAR-100. Fig. 1 illustrates the sensitivity to γ ($\gamma = 1.0$ means the baseline models). In 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. 3 and 4 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 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).

Table 3. 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 4. 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 [3]	5.83	4.55 (-1.28)	4.74	4.22 (-0.52)
SR [2]	6.80	4.75 (-2.05)	4.48	4.22 (-0.26)
SSD-GAN [1]	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)

F. 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 5 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.

G. 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 3 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

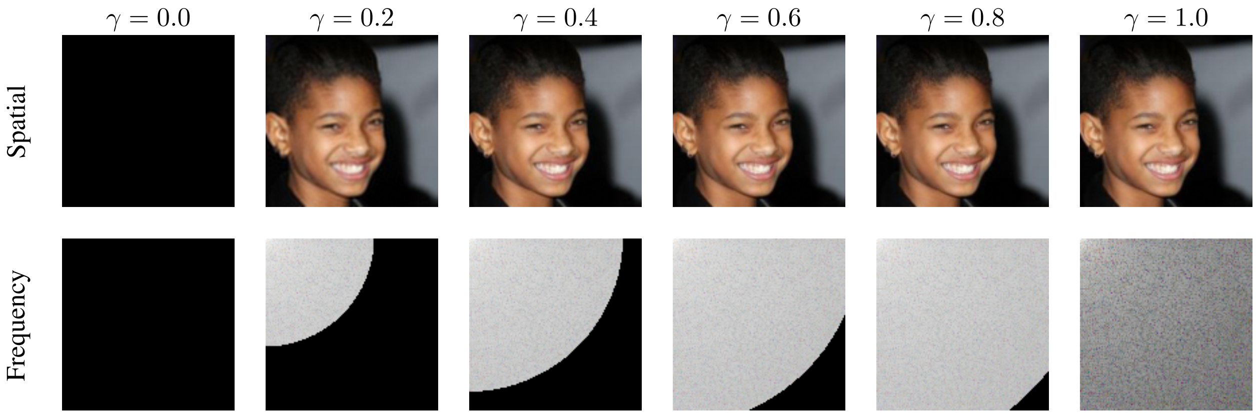

Figure 5. Effects of F-Drop on an CelebA sample

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 4 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.

H. Fake Detection

Similar to the evaluation presented in Frank *et al.* [3], 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 [3], 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. Mean accuracy of fake detection with linear binary classification (CIFAR-100)

	Spatial	Frequency
Baseline (SNGAN)	$90.4^{\pm1.2}$	$92.1^{\pm1.0}$
Binomial [3]	$95.7^{\pm0.3}$	$90.9^{\pm0.5}$
SR [2]	$88.2^{\pm1.2}$	$91.7^{\pm0.9}$
SSD-GAN [1]	$89.7^{\pm1.6}$	$93.2^{\pm0.6}$
F-Drop	$87.1^{\pm3.2}$	$89.8^{\pm2.4}$
F-Match	$81.0^{\pm1.2}$	$84.7^{\pm1.3}$
F-Drop&Match	$\mathbf{78.1}^{\pm\mathbf{0.7}}$	$\mathbf{83.1}^{\pm\mathbf{1.9}}$

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 of the main paper: applying a low-pass filter to GAN architectures may lead to difficulty in the training.

I. Additional Results of Single Fourier Attack

In Fig. 6, 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.

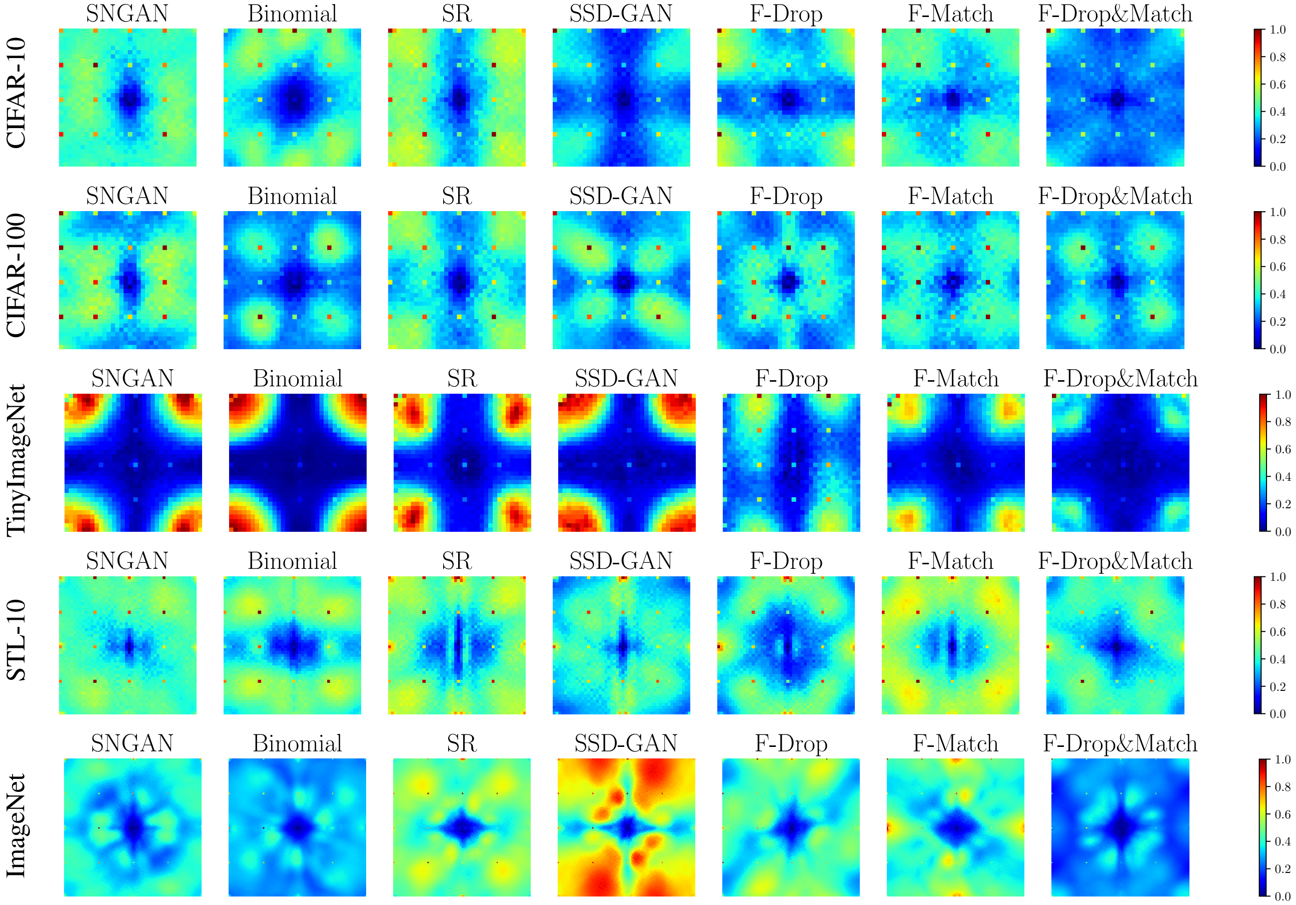

Figure 6. Sensitivity analysis by SFA [7] on multiple datasets

J. Additional Qualitative Results

We visualize the generated images from SNGAN and our F-Drop&Match for each dataset. Figure 7, 8, 9, 10, 11, 12, 13, 14, 15, and 16 illustrate 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 CIFAR-100 and textures of animal skins in STL-10.

References

[1] Yuanqi Chen, Ge Li, Cece Jin, Shan Liu, and Thomas Li. Ssd-gan: Measuring the realness in the spatial and spectral domains, 2020. 1, 3, 4

[2] 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, 3, 4

[3] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. 2020. 3, 4

[4] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014. 1

[5] 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. 1

[6] Jose Tribolet and Ronald Crochiere. Frequency domain coding of speech. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(5):512–530, 1979. 1

[7] 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, 5

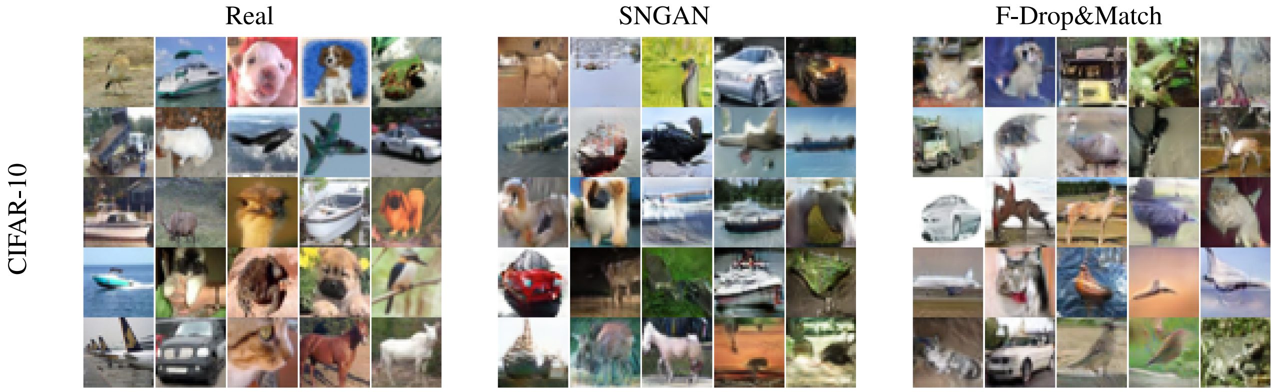

Figure 7. Generated images on CIFAR-10

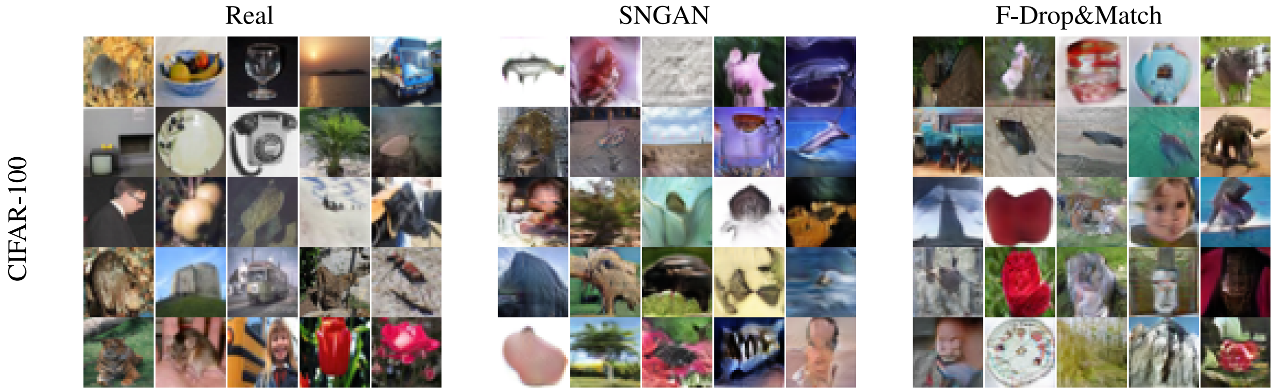

Figure 8. Generated images on CIFAR-100

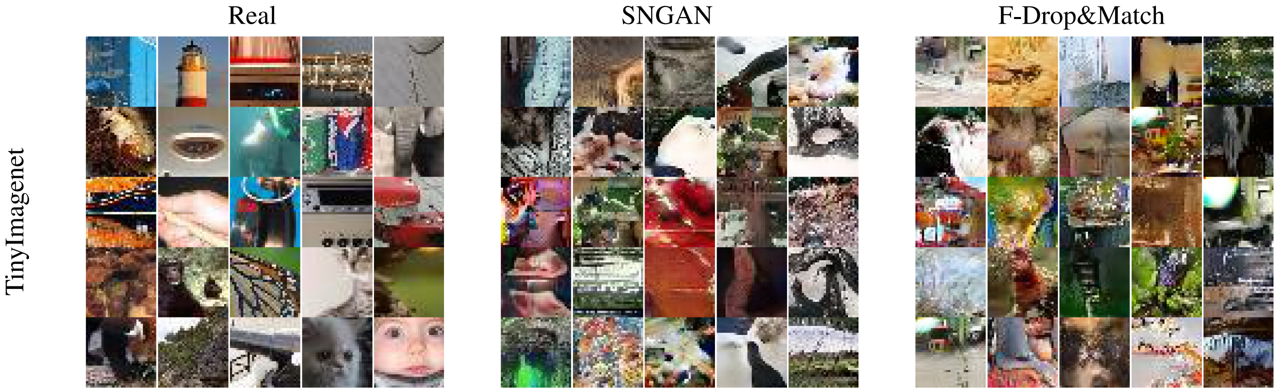

Figure 9. Generated images on TinyImageNet

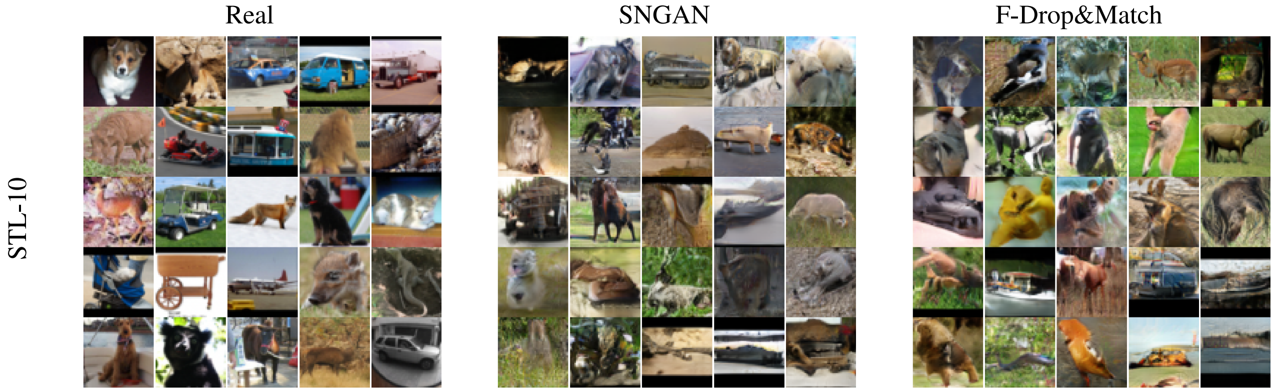

Figure 10. Generated images on STL-10

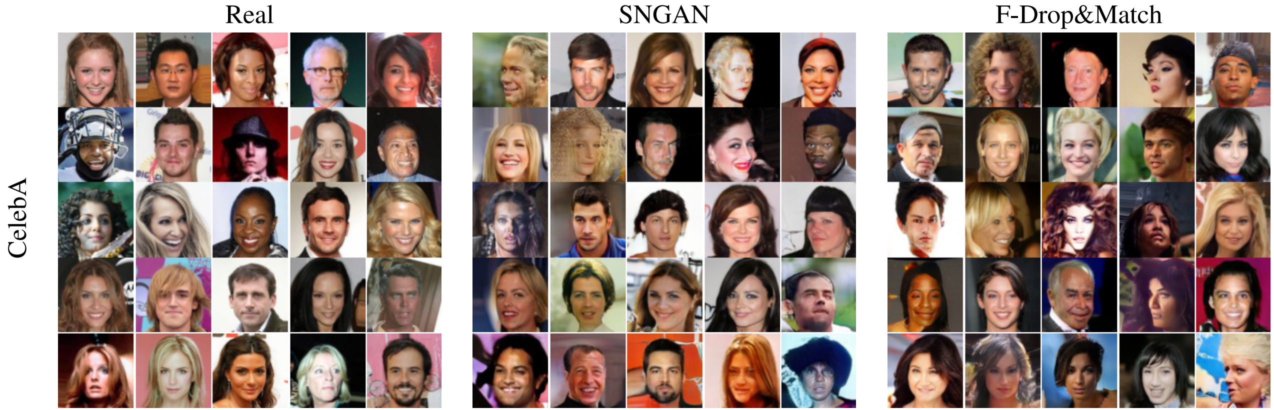

Figure 11. Generated images on CelebA

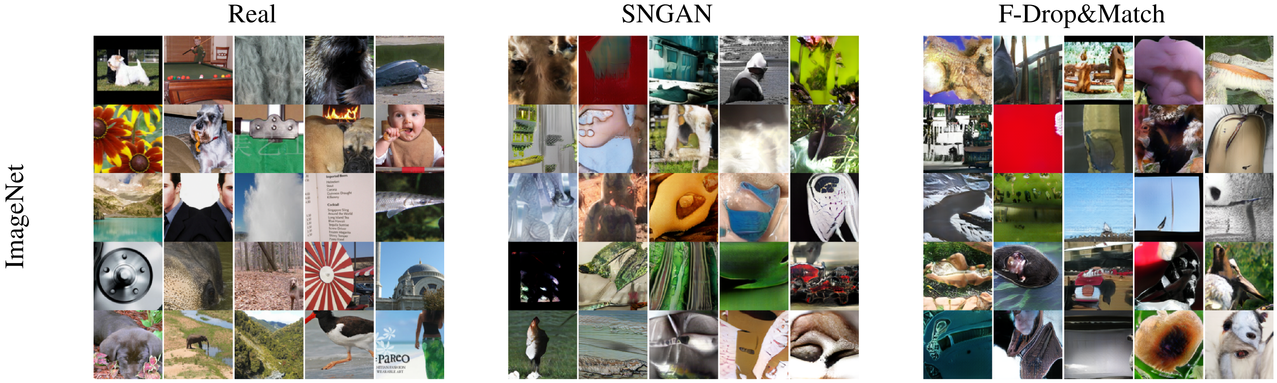

Figure 12. Generated images on ImageNet

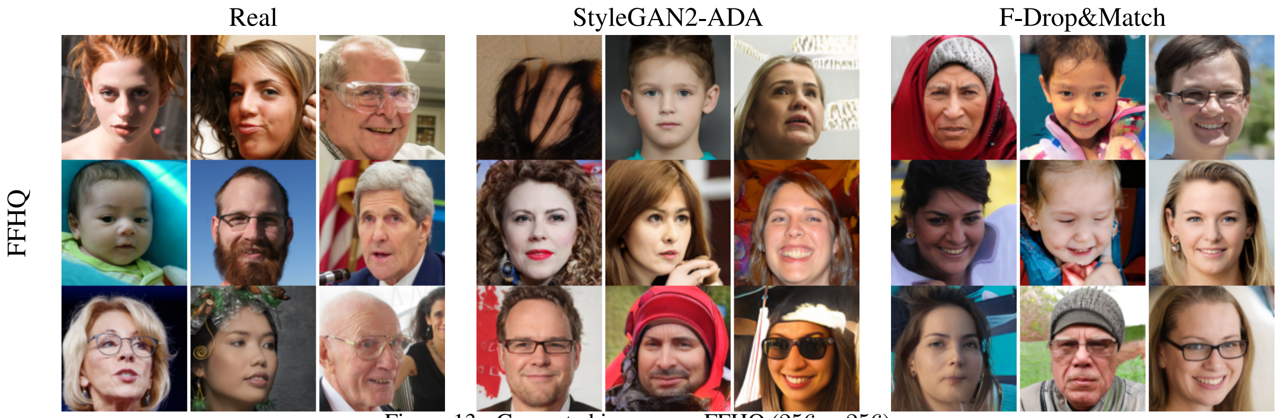

Figure 13. Generated images on FFHQ (256 × 256)

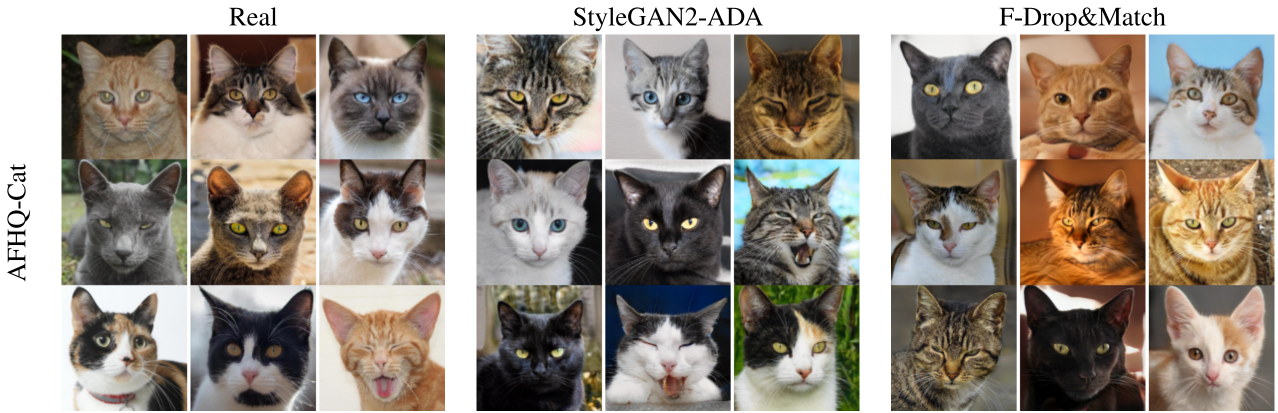

Figure 14. Generated images on AFHQ-Cat (512 × 512)

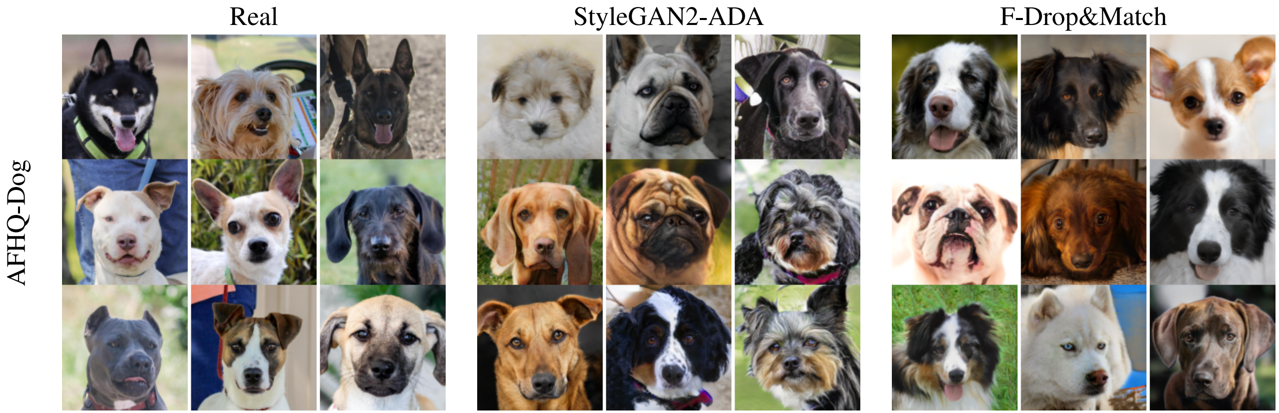

Figure 15. Generated images on AFHQ-Dog (512 × 512)

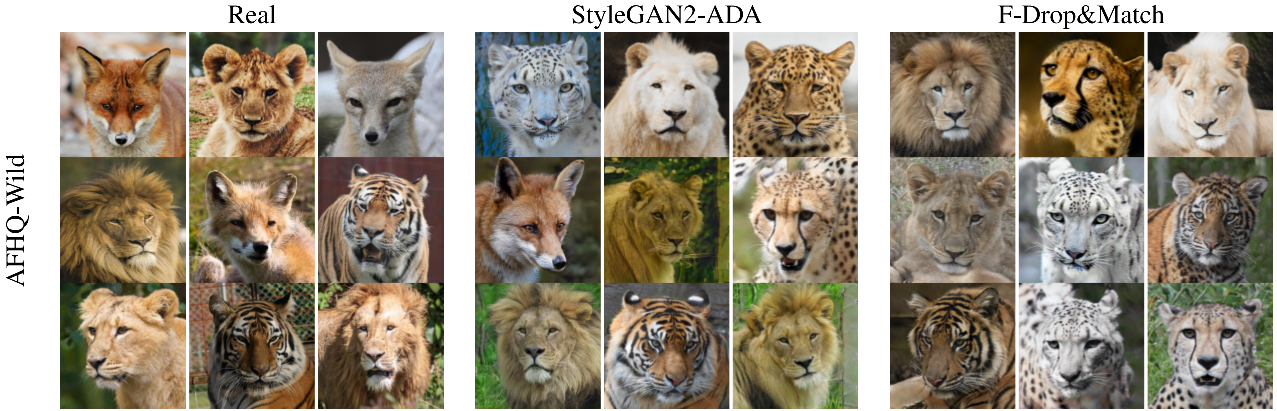

Figure 16. Generated images on AFHQ-Wild (512 × 512)