Focal Frequency Loss for Image Reconstruction and Synthesis

Supplementary Material

Liming Jiang[1] Bo Dai[1] Wayne Wu[2] Chen Change Loy[1]✉
[1]S-Lab, Nanyang Technological University [2]SenseTime Research
{liming002, bo.dai, ccloy}@ntu.edu.sg wuwenyan@sensetime.com

Abstract

This document provides supplementary information that is not elaborated in our main paper due to the space constraints: Section A shows some additional illustrations to explain our method further. Section B describes the implementation details in our experiments. Section C details our used datasets under diverse settings. Section D provides some studies on the variants of focal frequency loss. Section E presents additional results and analysis.

A. Additional Illustrations of Methodology

A.1. Spatial Frequency Visualization

After applying 2D discrete Fourier transform, an image is converted into its frequency representation and decomposed into orthogonal sine and cosine functions. The angular frequency of each sine and cosine function is decided by the frequency spectrum coordinate (u, v). The spatial frequency manifests as the 2D sinusoidal components in the image. The spectrum coordinate also represents the angled direction of a specific spatial frequency. As an intuitive view, we show some examples of the 2D sinusoidal components with specific spatial frequencies in Figure 1. It is observed that the angled direction and density (angular frequency) of the waves depend on the spectrum coordinate (u, v). Besides, the complex frequency value $F(u, v)$ can be regarded as the weight for each wave, and the weighted sum corresponds to the whole image in the spatial domain.

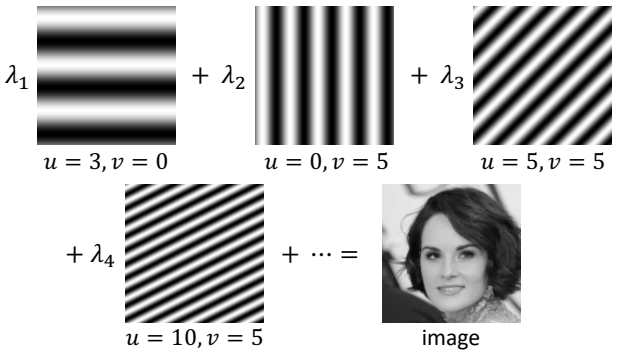

Figure 1. Two-dimensional sinusoidal components with specific spatial frequencies in an image. The angled direction and density (angular frequency) of the waves depend on the spectrum coordinate (u, v), and $F(u, v)$ can be seen as the weight for each wave.

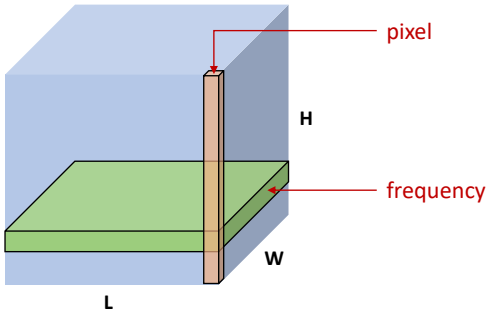

Figure 2. According to Figure 1, an image (gray-scale for simplicity) can be seen as a cube in a space, where its length (L) and width (W) dimensions correspond to the pixel domain, and the height (H) dimension corresponds to the frequency domain.

A.2. More Intuitive Illustration

To further explain the proposed focal frequency loss (FFL), we will provide a more intuitive illustration in this section. According to Figure 1, an image (gray-scale for simplicity) is the weighted sum of different spatial frequencies. We expand the accumulated frequencies into a new dimension, thus the image can be seen as a cube in a space. The length (L) and width (W) dimensions of the cube correspond to the pixel domain, and the height (H) dimension corresponds to the frequency domain, as shown in Figure 2. Therefore, a single pixel can be seen as the orange prism, and a specific frequency can be regarded as the green plane. It is observed that each frequency (*i.e.*, each coordinate value on the frequency spectrum) depends on all the image pixels. Due to the inherent bias of neural networks [24, 31], a model tends to eschew some frequency components that

are hard to synthesize, *i.e.*, hard frequencies, in the H dimension. Optimizing in the spatial domain (*i.e.*, in the L and W dimensions) hardly help the model locate these hard frequencies in the H dimension. Similarly, focusing on certain pixels (*e.g.*, orange prism) hardly help the model tackle the hard frequencies (*e.g.*, green plane). Intuitively, when directly optimizing in the H dimension (*i.e.*, explicitly using the frequency representation of the image in our method), the model can easily locate hard frequencies and in turn focus on them.

In Figure 2, it is noteworthy that each frequency also affects all the image pixels in the spatial domain. When FFL directly optimizes and adaptively focuses a model in the frequency domain, the frequency components in the H dimension will be reconstructed and synthesized better. Meanwhile, the general alignment and quality of all the image pixels in the L and W dimensions will be indirectly improved by FFL, thus boosting some pixel-based metrics (*e.g.*, PSNR and SSIM [29]) and ameliorating the image reconstruction and synthesis quality.

We wish to highlight that both the spatial-based loss and frequency-based loss are important since they consider different aspects and dimensions of an image, as illustrated in Figure 2. Hence, they are complementary and not replaceable. The proposed FFL is intending to complement existing spatial losses of different methods to improve reconstruction and synthesis quality further.

In fact, the actual situation of the frequency components in an image is much more complicated, which may be a higher-dimensional representation. The visualization in this section just provides a simple and intuitive illustration to help understand the proposed method in this paper.

B. Implementation Details

The code used for our experiments will be made publicly available. All the experiments are conducted on the NVIDIA Tesla V100 GPUs with 32 GB memory capacity.

B.1. Baseline Details

In this section, we will provide the implementation details of all the baselines in different image reconstruction and synthesis tasks. We select five representative methods from the two popular categories: autoencoder-based and GAN-based. Besides, we evaluate different network structures. Specifically, we explore the multilayer perceptron (MLP) network and the convolutional neural network (CNN). For CNN, the network details also vary, *e.g.*, with or without the skip connections. In addition, we consider various basic spatial domain losses, *e.g.*, MSE loss, L1 loss, GAN loss [5], perceptual loss [11], *etc.*, to test the ability of focal frequency loss to complement these losses.

Vanilla AE. Vanilla autoencoder [8] learns the image latent representation in an unsupervised manner, traditionally used for dimension reduction and feature learning. We employ vanilla AE in the image reconstruction task. The network is a simple 2-layer MLP with a hidden size of 256. We adopt ReLU activations (except the last layer using Tanh) and no norm layers. We use Adam [15] optimizer and set $\beta_1 = 0.9, \beta_2 = 0.999$. The learning rate is 0.001. Normal initialization (with mean 0.0 and standard deviation 0.02) is applied to all the networks of vanilla AE. The spatial loss is MSE loss. The models are trained on 1 GPU with a batch size of 128. We perform 200 epochs of training on DTD [2] and 20 epochs of training on CelebA [18].

VAE. Exploiting a reparameterization trick, the variational autoencoder [16] generates images by learning the latent representation in a probability distribution manner. We use VAE for image reconstruction and unconditional image synthesis. We employ CNN for VAE, with typical convolution and transposed convolution layers. Batch normalization [9] and Leaky ReLU (with a negative slope of 0.2, except the last layer using Tanh) are applied. Each convolution layer has a kernel size 4×4, stride 2, and zero-padding amount 1. In the encoder, the feature map resolution is halved after each convolution block. Images are down-sampled to 4×4, so the number of blocks depends on the input size (*e.g.*, if the input size is 64×64, there will be 4 blocks). After an input layer, the number of feature channels is 64. Then, the number of feature channels will double after each convolution block, while we set a maximum channel number to 512 to avoid using redundant parameters. We apply two linear layers to the encoded feature to learn μ and σ for the reparameterization. The latent size is 256. After reparameterization, an additional linear layer is used to adjust the feature to the original shape. In the decoder, the network structure is completely inverse to the encoder by replacing convolution layers with the transposed convolution layers. We use Adam [15] optimizer and set $\beta_1 = 0.9, \beta_2 = 0.999$. The learning rate is 0.001. Normal initialization (with mean 0.0 and standard deviation 0.02) is applied to all the networks of VAE. The spatial losses are MSE loss and KL divergence loss [16]. The models are trained on 1 GPU with a batch size of 128. We train our models for 20 epochs on CelebA [18] and 400 epochs on CelebA-HQ [12].

pix2pix. pix2pix [10] adopts conditional GAN [20] as a general-purpose solution to image-to-image translation with training pairs. We employ pix2pix for conditional image synthesis. The U-Net [25] generator is applied, which is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks. There are 8 skip connection blocks in the generator. The patch-based discriminator is used. Adam [15] optimizer is used with $\beta_1 = 0.5, \beta_2 = 0.999$. The learning rate is 0.0002. Normal initialization (with mean 0.0 and standard deviation 0.02) is applied to all the networks of pix2pix. The spatial losses are vanilla GAN loss [5] and L1 loss. The models are trained

on 1 GPU. We conduct 200 epochs of training on CMP Facades [23] with a batch size of 1. We train the models for 15 epochs on edges $\rightarrow$ shoes [32] with a batch size of 4. For other detailed network structures and parameters, we follow the original paper [10] and their released code.

SPADE. As a task-specific GAN-based method for semantic image synthesis (*i.e.*, synthesizing a photorealistic image from a semantic segmentation mask), SPADE [22] resizes the segmentation mask for modulating the activations in normalization layers by a learned affine transformation. The generator is built on a series of residual blocks [6] with the synchronized version of batch normalization. The multi-scale patch-based discriminator [28] with the instance normalization [27] is exploited. Besides, spectral normalization [21] is applied to all the convolutional layers in the generator and discriminator. Adam [15] optimizer is exploited with $\beta_1 = 0$, $\beta_2 = 0.9$. Two time-scale update rule [7] is applied, where the learning rates for the generator and the discriminator are 0.0001 and 0.0004, respectively. The spatial losses are hinge-based GAN loss [17, 21, 33], perceptual loss [11] calculated by VGG-19 [26] model, and feature matching loss [28]. The models are trained for 200 epochs on Cityscapes [3] and ADE20K [34] using 4 GPUs. The batch size is 32. For other detailed network structures and parameters, we follow the original paper [22] and their released code.

StyleGAN2. We further explore the potential of focal frequency loss on the state-of-the-art unconditional image synthesis method, StyleGAN2 [14]. We construct the StyleGAN2 baseline on top of its open-source official implementation. The mapping network consists of 8 fully connected layers. The dimensionality of both the input latent space and intermediate latent space is 512. The activation function is Leaky ReLU with a negative slope of 0.2. Several standard techniques in [12, 13] are applied, such as the exponential moving average of generator weights, mini-batch standard deviation layer at the end of the discriminator, equalized learning rate for all the trainable parameters, *etc*. Adam [15] optimizer is used with $\beta_1 = 0, \beta_2 = 0.99$. The spatial loss is non-saturating logistic loss [5, 14] with R_1 regularization [19]. All the models are trained with 8 V100 GPUs. The batch size is 64 for CelebA-HQ [12] (256×256) and 32 for the resolution of 1024×1024. For other detailed network structures and parameters, we follow the original paper [14] and their released code.

As for the relevant losses used for comparison, *i.e.*, perceptual loss [11] and spectral regularization [4], we follow all the details in their papers and released code.

B.2. Computational Cost

The computational cost of the proposed focal frequency loss (FFL) is negligible. Take pix2pix image-to-image translation on the CMP Facades dataset as an example. The average computational training time only increases from 0.064 to 0.067 seconds per iteration after applying FFL. The memory consumption increases from 3513 to 3515 MB. This cost test is conducted on 1 NVIDIA Tesla V100 GPU.

C. Dataset Details

In this section, we will provide detailed information about the seven datasets we explored. The datasets vary in types, sizes, and resolutions.

- **Describable Textures Dataset (DTD).** We use DTD provided by [2], which is an evolving collection of textural images in the wild, annotated with human-centric attributes. DTD contains texture images with special frequency patterns. We perform vanilla AE image reconstruction using this dataset, with $4,512$ images for training and $1,128$ images for testing. The original images are scaled and center cropped to 64×64.

- **CelebA.** CelebA [18] is a large-scale face attributes dataset covering large pose variations and background clutter. We conduct image reconstruction with vanilla AE and VAE on CelebA. Besides, we perform VAE unconditional image synthesis on CelebA. We use the cropped and aligned faces, which are more natural images. The training set contains $199,599$ images, and the test set has $3,000$ images. The images are resized and center cropped to 64×64.

- **CelebA-HQ.** CelebA-HQ is a higher-quality version of the CelebA dataset provided by [12]. The original resolution is 1024×1024. We perform VAE image reconstruction on this dataset. Besides, we study the unconditional image synthesis by VAE and StyleGAN2 using CelebA-HQ. The dataset is randomly split, yielding $27,000$ images for training and $3,000$ images for evaluation. All the cropped and aligned face images are uniformly resized to 256×256. For StyleGAN2, we also tried to synthesize images with a resolution of 1024×1024 besides 256×256.

- **CMP Facades.** For pix2pix image-to-image translation, we utilize the officially prepared CMP Facades [23] dataset. The facades are collected from different cities around the world with diverse architectural styles. CMP Facades contains architectural labels and photos, which is suitable for mask $\rightarrow$ image translation. The sizes of training and test sets are 400 and 106, respectively. The resolution is 256×256.

- **Edges $\rightarrow$ shoes.** We also exploit the officially prepared edges $\rightarrow$ shoes dataset for pix2pix image-to-image translation. The shoe images are from UT Zappos50K [32]. The shoes are centered on a white background. The edge maps are detected by HED [30]. The

numbers of images for training and testing are $49,825$ and 200, respectively. The image size is 256×256.

- **Cityscapes.** We use the Cityscapes [3] dataset for SPADE semantic image synthesis. Cityscapes dataset consists of street scene images that are mostly collected in Germany. The dataset provides instance-wise, dense pixel annotations of 30 classes. The training set has $2,975$ images, and the test set contains 500 images. The images are scaled to 512×256.

- **ADE20K.** ADE20K [34] dataset contains challenging in-the-wild images with fine annotations of 150 semantic classes. We also use ADE20K for SPADE semantic image synthesis, with $20,210$ images for training and $2,000$ images for evaluation. All the images are resized to 256×256.

D. Variant Studies

In our main paper, we mentioned that the exact form of the proposed focal frequency loss (FFL) is not crucial. In this section, we will provide some variants to extend and modify FFL. We will show some studies on these variants. For simplicity and intuitiveness, we revisit the vanilla AE image reconstruction task on CelebA. We report quantitative evaluation results for the variant studies. The visual results of variants are similar.

Several simple variants can be derived by adjusting the spectrum weight matrix parameter α. The parameter α controls how close the weight matrix values are, *i.e.*, how focused the model is. The larger α is, the model will be more focused on the hard frequencies, *i.e.*, the weight difference for easy and hard frequencies will be larger. For the experiments we present in our main paper, we set $\alpha = 1$ (we call the main version). The results are shown in Table 1. Applying the main version of FFL ($\alpha = 1$) shows better performance than the baseline without FFL in all the five metrics. If we set $\alpha = 2$, the quantitative results degrade from the main version, especially FID. This suggests that the model may be too focused on the hard frequencies while ignoring some important easy frequency information, albeit the results are still better than the baseline in most cases. When setting $\alpha = 0.5$, all the metric results are better than the baseline. The LPIPS and FID scores become better than the main version. The results of this variant are close to the main version of FFL. If we set $\alpha = 0.1$, the quantitative results degrade from the main version despite still better than the baseline. This indicates that the model may be too unfocused. For a trade-off, we select $\alpha = 1$ as the main version of FFL, while one may consider choosing other variants regarding the parameter α in certain tasks for the flexibility.

Besides, we study another category of variants, the patch-based focal frequency loss, where we crop an image into small patches so that the focused frequencies are at the patch level. We define the patch factor p as the number of patches on each edge. For instance, if $p = 2$, the image will be cropped into $2 \times 2 = 4$ patches. Obviously, using the original image without cropping it into patches, *i.e.*, the main version of FFL we defined before, corresponds to $p = 1$. The results are shown in Table 2. We note that $p = 1, 2, 4$ achieve close performance regarding the five evaluation metrics, all of which are much better than the baseline. However, if we set $p = 8$, the quantitative performance will degrade from the previous versions, especially FID. Although the results are still better than the baseline in most cases, this indicates that the patch size should not be too small. We simply choose $p = 1$ as the main version of FFL for our experiments in the main paper. However, the variant studies show that the patch-based focal frequency loss may contribute to an additional performance boost in certain cases. Thus, this may be another direction to extend and modify FFL.

Table 1. The PSNR (higher is better), SSIM (higher is better), LPIPS (lower is better), FID (lower is better) and LFD (lower is better) scores for the **variant studies** on the spectrum weight matrix **parameter** α for the focal frequency loss.

	PSNR↑	SSIM↑	LPIPS↓	FID↓	LFD↓
baseline	20.044	0.568	0.237	97.035	14.785
$\alpha = 1$ (main)	**21.703**	**0.642**	0.199	83.801	**14.403**
$\alpha = 2$	21.376	0.621	0.203	102.329	14.478
$\alpha = 0.5$	21.521	0.635	**0.197**	**82.561**	14.445
$\alpha = 0.1$	20.497	0.591	0.225	89.792	14.681

Table 2. The PSNR (higher is better), SSIM (higher is better), LPIPS (lower is better), FID (lower is better) and LFD (lower is better) scores for the **variant studies** on **patch-based** focal frequency loss. Patch factor p is the number of patches on each edge.

	PSNR↑	SSIM↑	LPIPS↓	FID↓	LFD↓
baseline	20.044	0.568	0.237	97.035	14.785
$p = 1$ (main)	21.703	0.642	0.199	**83.801**	14.403
$p = 2$	**21.836**	**0.648**	0.185	88.475	**14.372**
$p = 4$	21.752	0.643	**0.170**	90.612	14.392
$p = 8$	21.414	0.627	0.176	102.334	14.470

E. Additional Results and Analysis

E.1. Frequency Domain Gap

As mentioned in the main paper, we wish to improve the image reconstruction and synthesis quality by narrowing the frequency domain gap between the real and generated images using the proposed focal frequency loss (FFL). We have shown that the gaps between mini-batch average spectra of state-of-the-art StyleGAN2 are clearly mitigated by FFL. We will show some more examples of VAE image reconstruction on the CelebA [18] dataset and provide more analysis about the frequency domain gap in this section.

The results are shown in Figure 3. In the spatial domain, without applying FFL, the reconstructed faces are

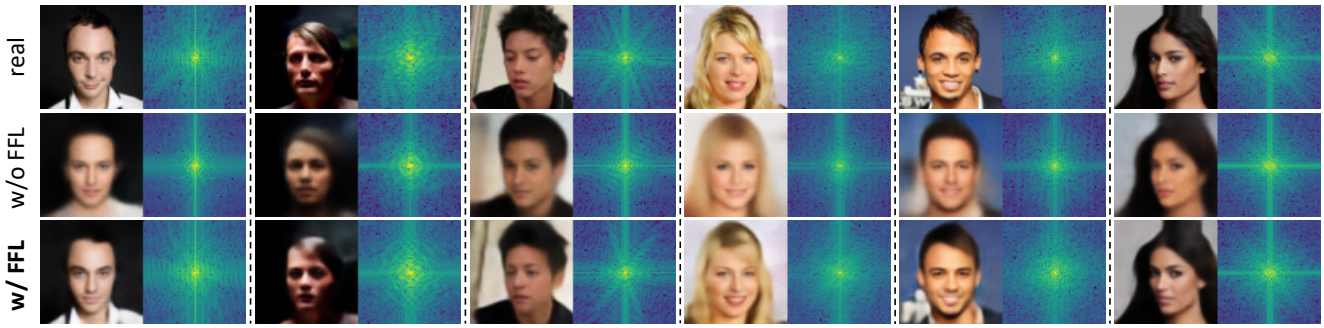

Figure 3. Frequency domain gaps are narrowed by the focal frequency loss (FFL) for VAE image reconstruction on CelebA.

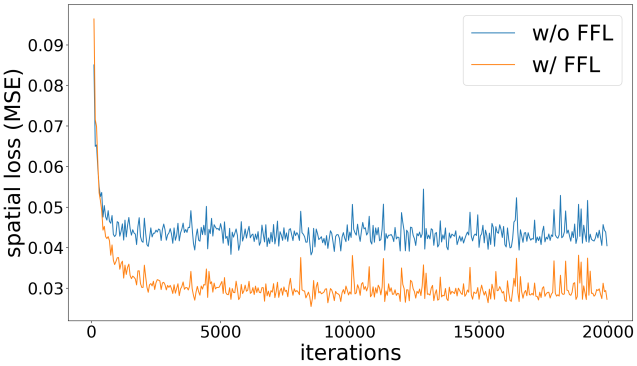

Figure 4. The spatial losses (MSE) with the same weight and random seed of the two training processes with/without focal frequency loss (FFL) for vanilla AE image reconstruction on CelebA. The spatial loss converges to a lower point with the help of FFL.

FID ↓	80.279	**74.359**	86.674	98.778	89.255	77.864
IS ↑	2.674	**2.804**	2.713	2.667	2.527	2.705

Figure 5. **Additional ablation studies** of each key component for the focal frequency loss (FFL), *i.e.*, frequency representation (freq), phase and amplitude (ampli) information, and dynamic spectrum weighting (focal) in the pix2pix image-to-image translation task on edges → shoes (256 × 256). The corresponding FID (lower is better) and IS (higher is better) scores are reported below the images.

blurry. This may be attributed to the reparameterization operation in the latent space between the encoder and decoder, which increases the difficulty for reconstruction. Trained with FFL, the VAE model can synthesize much clearer results, being closer to the ground truth real images. The perceptual quality is better after applying FFL. In the frequency domain, in line with our visualizations in the main paper, the VAE baseline without FFL bias to a limited spectrum region, losing high-frequency information (outer regions and corners). The frequency domain gaps are clearly narrowed after adopting FFL. The spectrum distribution becomes closer to the ground truth. Besides, some essential special spectrum patterns can be generated by applying FFL. This suggests the effectiveness of focal frequency loss to narrow the frequency domain gaps and ameliorate image quality further.

E.2. Training Loss

In the main paper, we have mentioned that the proposed focal frequency loss (FFL) is complementary to existing spatial losses, *e.g.*, MSE loss, to improve image reconstruction and synthesis quality. We further analyze the training loss in this section. We choose the vanilla AE image reconstruction task on CelebA [18] for simplicity. We plot the spatial losses with the same weight and random seed of the two training processes with/without FFL in Figure 4. It is readily observed that the spatial loss (MSE) converges to a lower point after applying FFL. This indicates that the model may converge to a better point with the help of FFL, in line with the better perceptual quality and quantitative performance we presented in our main paper.

E.3. Additional Ablation Studies

In the main paper, we provided the ablation studies of vanilla AE image reconstruction on CelebA for intuitiveness and simplicity, intending to study the importance of each key component for the proposed focal frequency loss (FFL) while reducing the influence of other factors, such as the adversarial loss. In this section, we provide the additional ablation studies on higher-resolution images with GAN. We show the studies of pix2pix [10] (*i.e.*, GAN-based method) image-to-image translation on edges → shoes (256 × 256) in Figure 5. The results are in line with the ablation studies in our main paper, further suggesting the importance of each key component for FFL.

E.4. Results on Non-Photorealistic Images

We further study the benefit of the proposed focal frequency loss (FFL) on non-photorealistic images. As an example, we provide the vanilla AE image reconstruction re-

Table 3. The PSNR (higher is better), SSIM (higher is better), LPIPS (lower is better), FID (lower is better) and LFD (lower is better) scores for the **vanilla AE image reconstruction** on **Danbooru2019 Portraits (Anime)** trained with/without the focal frequency loss (FFL).

Dataset	FFL	PSNR↑	SSIM↑	LPIPS↓	FID↓	LFD↓
Anime	w/o	19.885	0.575	0.294	193.342	14.822
(64 × 64)	w/	**20.657**	**0.628**	**0.267**	**184.443**	**14.644**

sults on Danbooru2019 Portraits [1] (Anime) in Table 3. Empirically, we observe that all the metrics can still be boosted by FFL. Our intuition is that FFL can also help generate non-photorealistic images since they still possess special frequency patterns that may be hard for a network to learn. FFL is adaptive for dealing with these frequencies.

E.5. Higher-Resolution Results on StyleGAN2

In Figure 6, we show some higher-resolution images synthesized by StyleGAN2 [14] trained with or without the proposed focal frequency loss (FFL) on CelebA-HQ (1024 × 1024). The truncation trick [13, 14] is not applied. Although the original StyleGAN2 (w/o FFL) generates plausible images in most cases, it sometimes produces tiny artifacts on the face (Row 2) and eyes (Row 3). The details on the teeth are missing in certain cases (Row 1). The synthesized images by StyleGAN2 with FFL (w/ FFL) are very photorealistic. Besides, StyleGAN2 achieves a better FID score after applying FFL, indicating that the quality of generated images becomes better with the help of FFL. More random sampled synthesized images without truncation are shown in Figure 7, and the examples with truncation using $\psi = 0.5$ [13, 14] are presented in Figure 8. It is observed that all the images generated by StyleGAN2 with FFL are with very high fidelity.

References

[1] Gwern Branwen, Anonymous, and Danbooru Community. Danbooru2019 Portraits: A large-scale anime head illustration dataset. https://www.gwern.net/Crops#danbooru2019-portraits. Accessed: 2021-04-10. 6

[2] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 2, 3

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 3, 4

[4] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: CNN based generative deep neural networks are failing to reproduce spectral distributions. In *CVPR*, 2020. 3

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2, 3

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[7] 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 *NeurIPS*, 2017. 3

[8] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313:504–507, 2006. 2

[9] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 2

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2, 3, 5

[11] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 2, 3

[12] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *arXiv preprint*, arXiv:1710.10196, 2017. 2, 3

[13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 3, 6, 9

[14] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020. 3, 6, 9

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*, arXiv:1412.6980, 2014. 2, 3

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint*, arXiv:1312.6114, 2013. 2

[17] Jae Hyun Lim and Jong Chul Ye. Geometric GAN. *arXiv preprint*, arXiv:1705.02894, 2017. 3

[18] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 2, 3, 4, 5

[19] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *ICML*, 2018. 3

[20] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint*, arXiv:1411.1784, 2014. 2

[21] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint*, arXiv:1802.05957, 2018. 3

[22] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 3

[23] Radim Šára Radim Tyleček. Spatial pattern templates for recognition of objects with regular structure. In *GCPR*, 2013. 3

[24] 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 *ICML*, 2019. 1

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*, arXiv:1409.1556, 2014. 3

[27] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint*, arXiv:1607.08022, 2016. 3

[28] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional GANs. In *CVPR*, 2018. 3

[29] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 13:600–612, 2004. 2

[30] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, 2015. 3

[31] 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. 1

[32] Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *CVPR*, 2014. 3

[33] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint*, arXiv:1805.08318, 2018. 3

[34] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017. 3, 4

Figure 6. Synthesis results (without truncation) of StyleGAN2 trained with/without the proposed FFL on CelebA-HQ (1024×1024). The model with FFL achieves the FID score of $\mathbf{3.374}$, outperforming the original StyleGAN2 without FFL of 3.733.

Figure 7. More random sampled images (without truncation) synthesized by StyleGAN2 trained with the proposed FFL on CelebA-HQ (1024×1024).

Figure 8. More random sampled images (with truncation applied using $\psi = 0.5$ [13, 14]) synthesized by StyleGAN2 trained with the proposed FFL on CelebA-HQ (1024×1024).