Supplementary Material for On the Frequency Bias of Generative Models

Katja Schwarz **Yiyi Liao** **Andreas Geiger**

Autonomous Vision Group
University of Tübingen and MPI for Intelligent Systems
{firstname.lastname}@uni-tuebingen.de

Abstract

In this **supplementary document**, we first provide implementation details on the testbeds for the generator and the discriminator in Section 1. In Section 2 and Section 3 we include additional analyses for the generator and discriminator, respectively. Section 4 comprises additional results for full GAN training. Lastly, we provide details on the datasets in Section 5. The **supplementary video** shows additional visualizations for the experiment in Fig. 1 of the main paper.

1 Implementation

1.1 Generator

The generator architecture is specified in Table 1 and is the same as the PGAN [8] generator except for a reduced number of channels. For the model, we base our implementation on https://github.com/rosinality/progressive-gan-pytorch. Following [8], we use a slope of 0.2 for the LeakyRelu and apply pixel normalization after every convolution. The first convolution uses a padding of 3, while the remaining convolutions retain the input resolution.
Note that upsampling by reshaping reduces the channel dimension by a factor of 4. Therefore, in this case, we multiply the output channels c_{out}, cf. Table 1, of each convolutional layer before the upsampling by 4. To keep a similar amount of total parameters, we further divide all channel dimensions by 1.5.
We train the model end-to-end using Adam optimizer [12] with a learning rate of 0.0001 and a batch size of 10. When training with both an L2-loss in image space and an L2-loss on the reduced spectrum, we weigh both losses equally. We train on a single NVIDIA Tesla V100-SXM2-32GB.
Teaser: For the experiment in Fig. 1b of the main paper, we train on a single image of resolution 128. We use the settings mentioned before but use a batch size of 1.

1.2 Discriminator

The discriminator architecture is specified in Table 2 and is the same as the PGAN [8] discriminator except for a reduced number of channels. For the model, we base our implementation on https://github.com/rosinality/progressive-gan-pytorch. Following [8], we use a slope of 0.2 for the LeakyRelu, and append the minibatch standard deviation after the last downsampling operation which increases c_{in} in Table 2 from 64 to 65. The last convolution uses no padding to reduce the spatial size from 4×4 to 1×1 before the linear layer, while the remaining convolutions retain the spatial size.
For the reconstruction task, we train a class-conditional discriminator with a single sample per class. Hence, the linear layer outputs n_c logits, where n_c is the number of classes, i.e., samples. In our experiments, we use $n_c = 10$. The generator is replaced by n_c learnable tensors which we optimize jointly with the discriminator. We train the model end-to-end using Adam optimizer [12]

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

Layer Type	Kernel Size	c_{in}	c_{out}	Activation	Normalization	Repetitions
Conv	4	64	64	LeakyRelu	PixelNorm	1
Conv	3	64	64	LeakyRelu	PixelNorm	1
Upsample	–	64	64	–	PixelNorm	$\times n$
Conv	3	64	64	LeakyRelu	PixelNorm	
Conv	3	64	64	LeakyRelu	PixelNorm	
Upsample	–	64	64	–	PixelNorm	$\times 1$
Conv	3	64	32	LeakyRelu	PixelNorm	
Conv	3	32	32	LeakyRelu	PixelNorm	
Upsample	–	32	32	–	PixelNorm	$\times 1$
Conv	3	32	16	LeakyRelu	PixelNorm	
Conv	3	16	16	LeakyRelu	PixelNorm	
Conv	1	16	3	–	PixelNorm	1

Table 1: **Architecture of the PGAN [8] Generator with Reduced Channels.** The value of n depends on the resolution of the data, e.g., $n = 2$ and $n = 3$ for resolution 64^2 and 128^2 pixels, respectively.

Layer Type	Kernel Size	c_{in}	c_{out}	Activation	Repetitions
Conv	1	3	16	LeakyRelu	1
Conv	3	16	32	LeakyRelu	1
Conv	3	32	32	LeakyRelu	1
Downsample	–	32	32	–	$\times 1$
Conv	3	32	64	LeakyRelu	
Conv	3	64	64	LeakyRelu	
Downsample	–	64	64	–	$\times n$
Conv	3	64	64	LeakyRelu	
Conv	3	64	64	LeakyRelu	
Downsample	–	64	64	–	$\times 1$
Conv	3	65	64	LeakyRelu	
Conv	4	64	64	LeakyRelu	
Linear	–	64	n_c	–	1

Table 2: **Architecture of the PGAN [8] Discriminator with Reduced Channels.** n_c denotes the number of classes. The value of n depends on the resolution of the data, e.g., $n = 2$ and $n = 3$ for resolution 64^2 and 128^2 pixels, respectively.

with a learning rate of 0.0001 for both the tensors and the discriminator, and a batch size of 10. To stabilize training we train with R1-regularization using a regularization strength of 10 and use an exponential moving average with decay 0.999 for the learnable tensors to produce the images. For all of our experiments, we ensure that training is stable by verifying that the discriminator converges to equilibrium, i.e., that the logits of the discriminator approach zero during training. For our testbed, we build on the framework from https://github.com/LMescheder/GAN_stability.git because it is intuitive and straightforward to work with. We train on a single NVIDIA Tesla V100-SXM2-32GB.
Teaser: For the experiment in Fig. 1c of the main paper, we train on a single image of resolution 128. We use the settings mentioned before but use a batch size of 1.

1.3 GAN

Our code framework for training the full GAN setting is based on the publicly available code for StyleGAN2-ada [9]: https://github.com/NVlabs/stylegan2-ada-pytorch.git because it is optimized for large-scale multi-GPU GAN training. We train PGAN on two NVIDIA Tesla V100-SXM2-32GB GPUs with a batch size of 256 and a learning rate of 0.002 for both the generator and the discriminator. The strength of the $R1$-regularizer is 10 and we use a minibatch size of 4 to compute the standard deviation within the batches. We do not train PGAN with adaptive discriminator augmentation and adhere to the training protocol from [11].
For finetuning StyleGAN2 we follow the training process of [11] using four V100-SXM2-32GB or four GEFORCE GTX 1080 Ti.

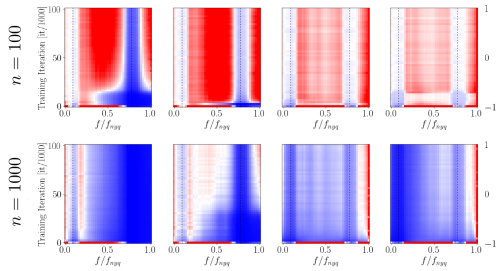


Figure 1: **Spectrum Error Evolution for Generators with Different Number of Training Samples on the Toyset.** Regardless of the number of training images, interpolation-based upsampling results in too little high-frequency content while zero insertion and reshaping struggle with checkerboard artifacts. The color corresponds to the relative error of the average predicted reduced spectrum wrt. the ground truth and is clipped at 1, i.e. when the relative error exceeds 100%.

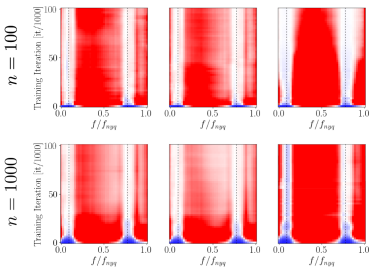


Figure 2: **Spectrum Error Evolution for Discriminators with Different Number of Training Samples on the Toyset.** Regardless of the number of training images, the discriminator shows no significant bias towards any frequency range. Instead, it generally struggles with frequencies of low magnitude. The color corresponds to the relative error of the average predicted reduced spectrum wrt. the ground truth and is clipped at 1, i.e. when the relative error exceeds 100%.

2 Additional Analysis of the Generator Testbed

2.1 Number of Images

We ablate how the number of training images impacts our generator testbed. Fig. 1 shows that the findings from the main paper remain consistent with a varying number of training images.

2.2 Low Magnitude Errors

In this section, we derive that an L2-loss penalizes errors with low magnitudes in the frequency domain less than errors with high magnitudes. Let us first consider a 1D-signal with discrete values x_n, $n = 0, \dots, N-1$. According to Parseval's theorem, for the discrete Fourier transform $\mathcal{F}$ it holds that

$$\sum_{n=0}^{N-1} |x_n|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |\mathcal{F}[\mathbf{x}]_k|^2 \tag{1}$$

where $|\mathcal{F}[\mathbf{x}]_k|^2$ is the magnitude at frequency k. For an L2-loss on a predicted signal with values x_n^{pred} and a target signal with values x_n^{tgt} we obtain

$$\sum_{n=0}^{N-1} \left| x_n^{pred} - x_n^{tgt} \right|^2 = \sum_{n=0}^{N-1} \left| \left[\mathbf{x}^{pred} - \mathbf{x}^{tgt} \right]_n \right|^2 \tag{2}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} \left| \mathcal{F} \left[\mathbf{x}^{pred} - \mathbf{x}^{tgt} \right]_k \right|^2 \tag{3}$$

Let us now consider some frequency k_0. When the error at k_0 has a low magnitude, then $|\mathcal{F} \left[\mathbf{x}^{pred} - \mathbf{x}^{tgt} \right]_{k_0}|^2$ is small and therefore contributes only slightly to the sum in Eq. (3).
For a gray scale image with pixel values x_{ij}, $i = 0, \dots, H-1$, $j = 0, \dots, W-1$, this follows similarly from Parseval's theorem in 2D

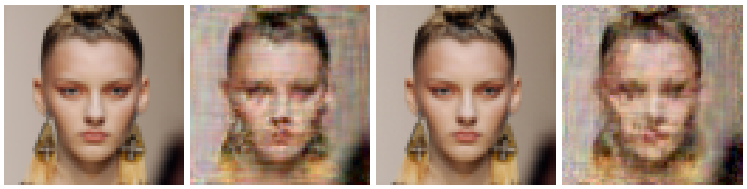
$$\sum_{i=0}^{H-1} \sum_{j=0}^{W-1} |x_{ij}|^2 = \frac{1}{HW} \sum_{k=0}^{H-1} \sum_{l=0}^{W-1} |\mathcal{F}[\mathbf{x}]_{kl}|^2 \tag{4}$$

	AvgPool	BlurPool	Stride
SD	31.6	26.2	27.7
SD-FT	53.9	53.0	46.3
SD-CNN-FT	27.5	24.9	26.0

Table 3: **PSNR** for different spectral discriminators on CelebA at resolution 64^2 pixels.

	Original		Wavelet		F-Mining		SD	
	Acc	FID	Acc	FID	Acc	FID	Acc	FID
SNGAN	75	78.0	59	59.2	65	80.6	58	63.0

Table 4: **Spectral Classification Accuracy and FID** for SNGAN on Cats128 with discriminators on different input domains.



(a) GT (b) SD (c) SD-FT (d) SD-CNN-FT

Figure 3: **Reconstruction Guided by the Discriminator** with BlurPool downsampling for different spectral discriminators on CelebA at resolution 64^2 pixels.

3 Additional Analysis of the Discriminator Testbed

3.1 Number of Images

We ablate how the number of training images impacts our discriminator testbed. Fig. 2 shows that the findings from the main paper remain consistent with a varying number of training images.

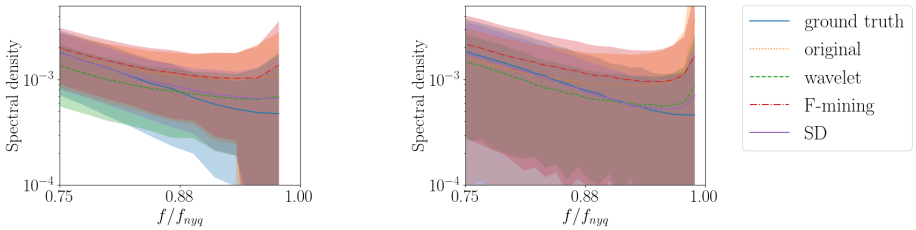
3.2 Penalizing the Spectrum

Jung et al. [7] propose an additional discriminator on the reduced spectrum (SD). However, penalizing the reduced spectrum might not be sufficient for improving the training signal alone because the spectrum computation and the azimuthal integration discard information, see Section 4 of the main paper. Therefore, we investigate if using the full Fourier transform instead of the reduced spectrum can improve performance. We compare two different additional discriminators to the version from [7]: The first is an MLP that operates on the flattened 2D Fourier transform of the image and therefore uses neither convolutions nor downsampling (SD-FT). Since this is only feasible for images with a small resolution, we also investigate a convolutional spectral discriminator (SD-CNN-FT) with the same architecture as the spatial discriminator. We weigh the spatial and spectral discriminator equally as done in [7]. Further, we ensure that the models of all approaches have a similar number of parameters. Note that using an MLP on the flattened 2D Fourier transform increases the number of parameters from $\sim$ 300k to $\sim$ 900k. Hence, we increase the channel dimensions of the discriminator for both SD and SD-CNN-FT to obtain models of comparable size.
The PSNR values in Table 3 show a significant improvement for all downsampling techniques for SD-FT. However, with the convolutional variant, SD-CNN-FT, the PSNR does not improve wrt. SD. This is consistent with the qualitative results for BlurPool downsampling in Fig. 3. Even with the increased amount of parameters, SD cannot remove the downsampling artifacts. While SD-FT closely reconstructs the ground truth, it can only be used with low-resolution images due to its fully connected architecture. Unfortunately, naïvely applying a convolutional architecture on the full spectrum also suffers from artifacts in the reconstructions due to downsampling. These observations suggest that integrating frequency domain supervision effectively and efficiently remains an open question in a GAN setting.

4 Additional Analysis of Full GAN Training

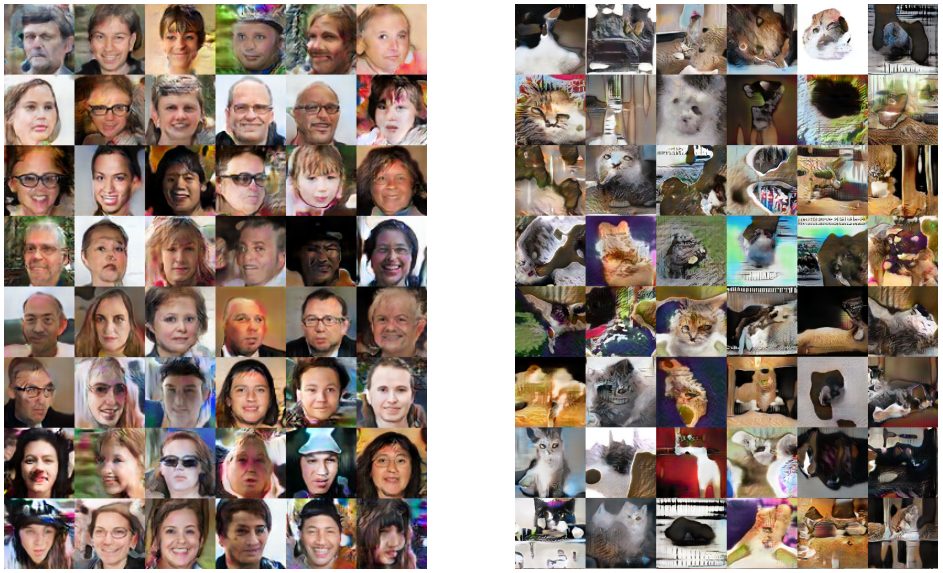
4.1 PGAN

Discriminators on Different Input Domains: For our ablation on different input domains for the discriminator, Fig. 4 shows spectrum plots corresponding to Table 3 in the main paper. In agreement with the observations in [2], F-mining alters the spectral statistics at the highest frequencies only



(a) FFHQ64 (b) Cats128

Figure 4: **Reduced Spectrum for PGAN with Discriminators on Different Input Domains.** We plot the mean and standard deviation of the reduced spectrum above $0.75 f_{nyq}$.



(a) FFHQ64 (b) Cats128

Figure 5: **Random Samples from PGAN with Reduced Channels.**

slightly. Wavelets reduce the peak at the highest frequencies but also predict too few frequencies below $0.88 f_{nyq}$. Similar to wavelets, the additional spectral discriminator (SD) reduces the peak at the highest frequencies but matches the spectral statistics below $0.88 f_{nyq}$ more closely. These observations are consistent across both datasets and support the reported classification accuracy in Table 3 of the main paper.

Qualitative Results for PGAN: In Fig. 5, we include some qualitative results for our smaller version of PGAN in the original setting. While not completely photo-realistic, the model can still reproduce characteristic features of the ground truth data. For our analysis, this is sufficient because the focus of these experiments is not on image fidelity but to study the spectral properties of both the generator and the discriminator in combination.

4.2 SNGAN

In this section, we consider different discriminators for SNGAN [14] to investigate if our analysis from Section 4 in the main paper is consistent across architectures. We base our framework on the official implementation of [3], https://github.com/cyq373/SSD-GAN.git. Similar to Section 4 in the main paper, we compare an additional spectral discriminator (SD) [7], hard example mining in the frequency domain (F-Mining) [3], and training in wavelet space (Wavelet) [6]. Consistent with our analysis on PGAN [8], Fig. 6 shows that the additional spectral discriminator is the most effective to reduce the spectral discrepancies. However, interestingly, the corresponding classification accuracies on the reduced spectra in Table 4 are similar for wavelets and the additional spectral discriminator. This suggests that training the spectral classifier on the fitted decay parameters, as proposed in [5], can occasionally produce overly optimistic results with an accuracy closer to chance, i.e., 50%. However,

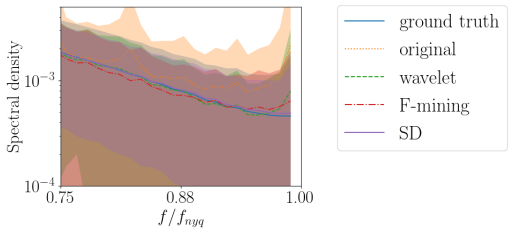
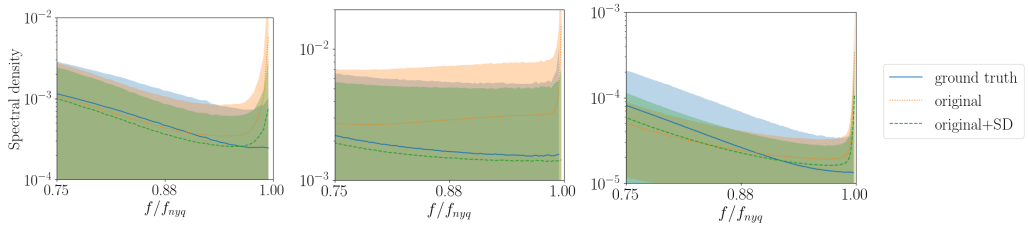


Figure 6: **Reduced Spectrum for SNGAN with Discriminators on Different Input Domains** on Cats128. We plot the mean and standard deviation of the reduced spectrum above $0.75 f_{nyq}$.



(a) Cats256 (b) AFHQ Dog (c) FFHQ

Figure 7: **Reduced Spectrum for StyleGAN2.** We plot the mean and standard deviation of the reduced spectrum above $0.75 f_{nyq}$. While the spectral discriminator removes the high-frequency peak on AFHQ Dog, the spectra for the remaining datasets retain an elevated amount of high frequencies.

for all results in the main paper the classification accuracy is consistent with the qualitative alignment of the reduced spectra, see Fig. 4, Fig. 7, and Fig. 7 in the main paper. To adhere to the same metric as [1, 5], we hence decide to report the classification accuracy on the fitted decay parameters.
While the FID in Table 4 is mixed for the different approaches, all values are lower than the corresponding values for PGAN in Table 3 of the main paper. This is expected, as our version of PGAN with the reduced channels has significantly fewer parameters than the original SNGAN.

4.3 StyleGAN2

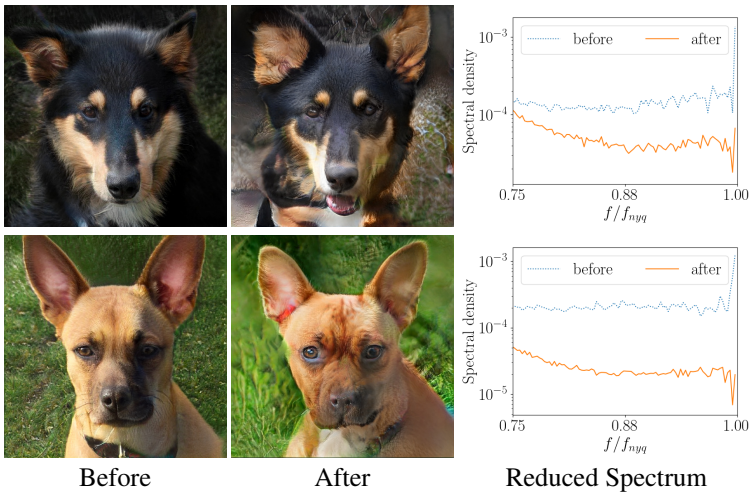
For finetuning StyleGAN2 [11] with the spectral discriminator [7], Fig. 7 shows spectrum plots corresponding to Table 5 in the main paper. While the spectral discriminator largely improves the spectral statistics on AFHQ Dog, it cannot fully resolve the peak at the highest frequencies for the remaining datasets. This also reflects in the high accuracy of the spectral classifier on these datasets in Table 5 of the main paper. Fig. 8 shows qualitative results before and after finetuning on AFHQ dog and FFHQ. For AFHQ dog, finetuning prevailingly changes the background to correct the spectral statistics but qualitatively this reduces the image fidelity. This supports the results in Table 5 in the main paper, where the accuracy of the spectral classifier is reduced but FID becomes worse. The finetuned images on FFHQ have no background artifacts and have a similar FID as the original images. However, the spectral statistics improve only slightly, see Fig. 7. This again suggests that the reduced spectrum might not contain enough information to correct both the spectral statistics and image fidelity.

5 Datasets

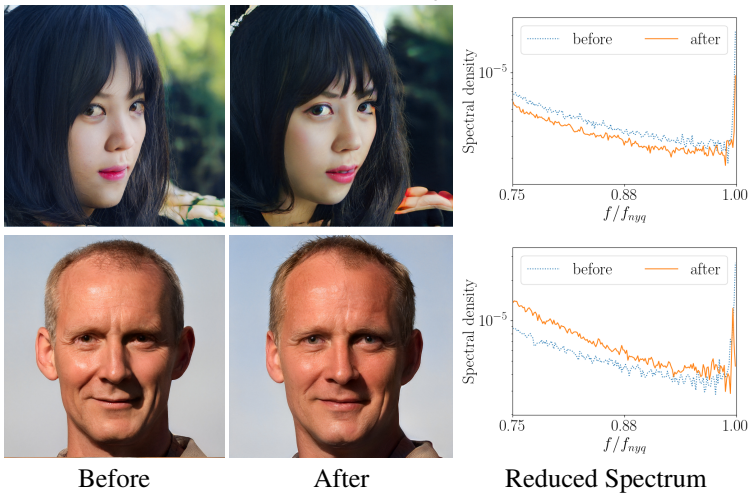
Toyset: We will include the scripts for generating our Toyset in our code release upon acceptance.
Licenses: The datasets used in this paper, CelebA [13], FFHQ [10] (Creative Commons BY-NC-SA 4.0), LSUN Cats [15], and AFHQ [4] (Creative Commons BY-NC 4.0) are available for non-commercial research purposes and are therefore suitable for our work.

References

[1] K. Chandrasegaran, N. Tran, and N. Cheung. A closer look at fourier spectrum discrepancies for cnn-generated images detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 6



(a) AFHQ Dog



(b) FFHQ

Figure 8: **Samples from StyleGAN2** and their spectra before and after finetuning with an additional discriminator on the reduced spectrum. (a) The peak at the highest frequencies is removed but image quality degrades. (b) The image fidelity remains high but the high-frequency artifacts also persist.

[2] K. Chen, R. Oldja, N. Smolyanskiy, S. Birchfield, A. Popov, D. Wehr, I. Eden, and J. Pehserl. Mvlidarnet: Real-time multi-class scene understanding for autonomous driving using multiple views. *arXiv.org*, 2006.05518, 2020. 4

[3] Y. Chen, G. Li, C. Jin, S. Liu, and T. Li. SSD-GAN: measuring the realness in the spatial and spectral domains. In *Proc. of the Conf. on Artificial Intelligence (AAAI)*, 2021. 5

[4] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[5] T. Dzanic, K. Shah, and F. D. Witherden. Fourier spectrum discrepancies in deep network generated images. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. 5, 6

[6] R. Gal, D. Cohen, A. Bermano, and D. Cohen-Or. SWAGAN: A style-based wavelet-driven generative model. *CoRR*, 2102.06108, 2021. 5

[7] S. Jung and M. Keuper. Spectral distribution aware image generation. In *Proc. of the Conf. on Artificial Intelligence (AAAI)*, 2021. 4, 5, 6

[8] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2018. 1, 2, 5

[9] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. 2

[10] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[11] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of StyleGAN. 2020. 2, 6

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2015. 1

[13] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2015. 6

[14] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2018. 5

[15] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv.org*, 1506.03365, 2015. 6