Supplemental Material: A Style-Based Generator Architecture for Generative Adversarial Networks

Tero Karras
NVIDIA
tkarras@nvidia.com

Samuli Laine
NVIDIA
slaine@nvidia.com

Timo Aila
NVIDIA
taila@nvidia.com

1. Hyperparameters and training details

We build upon the official TensorFlow [1] implementation of Progressive GANs by Karras et al. [4], from which we inherit most of the training details.[1] This original setup corresponds to configuration A in Table 1 of the paper. In particular, we use the same discriminator architecture, resolution-dependent minibatch sizes, Adam [5] hyperparameters, and exponential moving average of the generator. We enable mirror augmentation for CelebA-HQ and FFHQ, but disable it for LSUN. Our training time is approximately one week on an NVIDIA DGX-1 with 8 Tesla V100 GPUs.

For our improved baseline (B in Table 1), we make several modifications to improve the overall result quality. We replace the nearest-neighbor up/downsampling in both networks with bilinear sampling, which we implement by low-pass filtering the activations with a separable 2^{nd} order binomial filter after each upsampling layer and before each downsampling layer [12]. We implement progressive growing the same way as Karras et al. [4], but we start from 8^2 images instead of 4^2. For the FFHQ dataset, we switch from WGAN-GP to the non-saturating loss [2] with R_1 regularization [7] using $\gamma = 10$. With R_1 we found that the FID scores keep decreasing for considerably longer than with WGAN-GP, and we thus increase the training time from 12M to 25M images. We use the same learning rates as Karras et al. [4] for FFHQ, but we found that setting the learning rate to 0.002 instead of 0.003 for 512^2 and 1024^2 leads to better stability with CelebA-HQ.

For our style-based generator (F in Table 1), we use leaky ReLU [6] with $\alpha = 0.2$ and equalized learning rate [4] for all layers. We use the same feature map counts in our convolution layers as Karras et al. [4]. Our mapping network consists of 8 fully-connected layers, and the dimensionality of all input and output activations — including $\mathbf{z}$ and $\mathbf{w}$ — is 512. We found that increasing the depth of the mapping network tends to make the training unstable with high learning rates. We thus reduce the learning rate by two orders of magnitude for the mapping network, i.e., $\lambda' = 0.01 \cdot \lambda$. We initialize all weights of the convolutional, fully-connected, and affine transform layers using $\mathcal{N}(0, 1)$. The constant input in synthesis network is initialized to one. The biases and noise scaling factors are initialized to zero, except for the biases associated with $\mathbf{y}_s$ that we initialize to one.

[1] https://github.com/tkarras/progressive_growing_of_gans

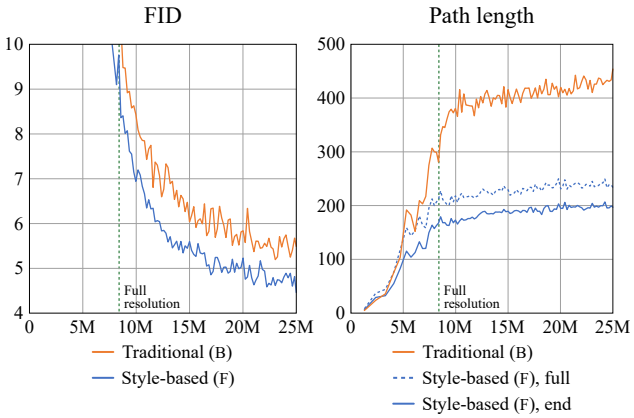

Figure 1. FID and perceptual path length metrics over the course of training in our configurations B and F using the FFHQ dataset. Horizontal axis denotes the number of training images seen by the discriminator. The dashed vertical line at 8.4M images marks the point when training has progressed to full 1024^2 resolution. On the right, we show only one curve for the traditional generator's path length measurements, because there is no discernible difference between full-path and endpoint sampling in $\mathcal{Z}$.

The classifiers used by our separability metric (Section 4.2 of the paper) have the same architecture as our discriminator except that minibatch standard deviation [4] is disabled. We use the learning rate of 10^{-3}, minibatch size of 8, Adam optimizer, and training length of 150,000 images. The classifiers are trained independently of generators, and the same 40 classifiers, one for each CelebA attribute, are used for measuring the separability metric for all generators. We will release the pre-trained classifier networks so that our measurements can be reproduced.

We do not use batch normalization [3], spectral normalization [8], attention mechanisms [11], dropout [9], or pixelwise feature vector normalization [4] in our networks.

Figure 2. Uncurated set of images produced by our style-based generator (config F) with the LSUN BEDROOM dataset at 256^2. FID computed for 50K images was 2.65.

Figure 3. Uncurated set of images produced by our style-based generator (config F) with the LSUN CAR dataset at 512×384. FID computed for 50K images was 3.27.

2. Training convergence

Figure 1 shows how the FID and perceptual path length metrics evolve during the training of our configurations B and F with the FFHQ dataset. With R_1 regularization active in both configurations, FID continues to slowly decrease as the training progresses, motivating our choice to increase the training time from 12M images to 25M images. Even when the training has reached the full 1024^2 resolution, the slowly rising path lengths indicate that the improvements in FID come at the cost of a more entangled representation. Considering future work, it is an interesting question whether this is unavoidable, or if it were possible to encourage shorter path lengths without compromising the convergence of FID.

3. Other datasets

Figures 2, 3, and 4 show an uncurated set of results for LSUN [10] BEDROOM, CARS, and CATS, respectively. In these images we used the truncation trick from Appendix Bwith $\psi = 0.7$ for resolutions $4^2 - 32^2$. The accompanying video provides results for style mixing and stochastic variation tests. As can be seen therein, in case of BEDROOM the coarse styles basically control the viewpoint of the camera, middle styles select the particular furniture, and fine styles deal with colors and smaller details of materials. In CARS the effects are roughly similar. Stochastic variation affects primarily the fabrics in BEDROOM, backgrounds and headlamps in CARS, and fur, background, and interestingly, the positioning of paws in CATS. Somewhat surprisingly the wheels of a car never seem to rotate based on stochastic inputs.

These datasets were trained using the same setup as FFHQ for the duration of 70M images for BEDROOM and CATS, and 46M for CARS. We suspect that the results for BEDROOM are starting to approach the limits of the training data, as in many images the most objectionable issues are the severe compression artifacts that have been inherited from the low-quality training data. CARS has much higher quality training data that also allows higher spatial resolution (512×384 instead of 256^2), and CATS continues to be a difficult dataset due to the high intrinsic variation in poses, zoom levels, and backgrounds.

Figure 4. Uncurated set of images produced by our style-based generator (config F) with the LSUN CAT dataset at 256^2. FID computed for 50K images was 8.53.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: a system for large-scale machine learning. In *Proc. 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, pages 265–283, 2016. 1

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. In *NIPS*, 2014. 1

[3] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. 1

[4] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017. 1

[5] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[6] A. L. Maas, A. Y. Hannun, and A. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. International Conference on Machine Learning (ICML)*, volume 30, 2013. 1

[7] L. Mescheder, A. Geiger, and S. Nowozin. Which training methods for GANs do actually converge? *CoRR*, abs/1801.04406, 2018. 1

[8] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *CoRR*, abs/1802.05957, 2018. 1

[9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. 1

[10] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015. 2

[11] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. *CoRR*, abs/1805.08318, 2018. 1

[12] R. Zhang. Making convolutional networks shift-invariant again, 2019. 1