Supplementary Material: Image Generation From Small Datasets via Batch Statistics Adaptation

Atsuhiro Noguchi[1] and Tatsuya Harada[1,2]

[1]The University of Tokyo, [2]RIKEN

A. Experimental detail

A.1. Anime face sampling

The Anime face dataset has a very high diversity in texture and facial shape and pose, which makes the dataset notably sparse when the dataset size is small. Therefore, we sampled images that have similar Gram matrix [2], which is known to control the style information, to limit the textural diversity. This made the problem easier.

A.2. Model selection

We used SNGAN [3] for unconditional GAN model, and SNGAN projection [3, 4] and BigGAN [1] for conditional GAN model. We used SNGAN-128 in the official SNGAN implementation[1] and our reimplementation of BigGAN-256 with the official pretrained weight[2]. We used VGG16 [5] trained on ImageNet for perceptual loss. We used layer ' conv1_1', 'conv1_2', 'conv2_1', 'conv2_2', 'conv3_1', 'conv3_2', 'conv3_3', 'conv4_1', 'conv4_2', and 'conv4_3' for the perceptual loss. During training, for SNGAN, we updated scale and shift parameters of all conditional batch normalization layers and the fully connected layer in the generator. For BigGAN, statistics for the fully connected layer and all parameters to calculate batch statistics were updated.

A.3. Training settings

In this subsection, we describe the training setting for the experiments. Some experiments are trained with a different setting from the ones bellow. For details, please refer to our implementation.

A.3.1 For SNGAN

All models were trained for 3,000 to 4,000 training iterations with batchsize 25. We used Adam optimizer with initial learning rate 0.1 for datasize 25, 0.06 for datasize 50, 0.03 for datasize 100, and 0.02 for datasize 500. We used $\lambda_C = 0.001$, $\lambda_z = 0.2$, and $0 \leq \lambda_{\gamma,\beta} \leq 0.02$. For comparison experiments, we used Adam optimizer with learning rate 0.001 for encoders, and 0.0001 for the other GAN models. We used the L1 norm for the perceptual loss. It takes an hour on a single Nvidia P100 GPU for 3,000 training iterations.

A.3.2 For BigGAN

All models were trained for 6000 to 10000 training iterations with batchsize 16. We used Adam optimizer with learning rate 0.001 for the parameters of class embeddings, 0.0005 for other statistics parameters, and 0.05 for latent vectors. We used $\lambda_C^l = 0.1/\sum_i \frac{1}{c_l h_l w_l}||C^{(l)}(x_i) - C^{(l)}(G(z_i+\epsilon))||_2$ instead of constant value, $\lambda_z = 0.2$, and $\lambda_{\gamma,\beta} = 0$. We found that with such λ_C, clearer images are generated. We used L2 norm for the perceptual loss. It takes 3 hours on 4 Nvidia P100 GPUs for 10,000 training iterations.

B. Relationship between the scale and shift and activation rate

In Section 3, we stated that changing γ and β is equivalent to controlling the activation. In this section, we investigated the relationship between the scale γ and shift β and the activation rate of each filter in SNGAN projection trained on ImageNet. For the "blenheim spaniel" class, in Figure A, we show a plot of the relationship between γ and β during batch normalization and the activation rate of the output of the activation function in each layer, where each column indicates the results of the first conditional batch normalization in each residual block in the generator, where each point represents each filter. For all cases, a positive correlation exists between γ and β and the activation rate. Therefore, it can be stated that changing these parameters is equivalent to filter selection.

[1]github.com/pfnet-research/sngan_projection
[2]tfhub.dev/deepmind/biggan-256/2

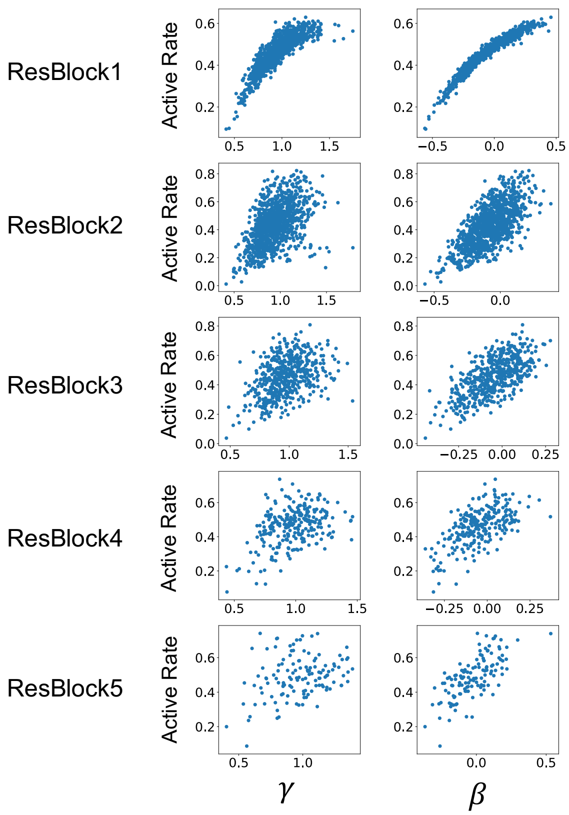

Figure A: Relationship between the rate of active kernel and γ and β.

C. Generated samples from Transfer GAN and "Update all"

In this section, we show the generative results from Transfer GAN [6] and "Update all" for each data size, which transfers prior knowledge of generative models similarly to our method. We used human face, anime face, and flower images for training and chose 25, 50, 100, and 500 as the data size. We did not test 500 for flower dataset because the dataset has only 251 images. We stopped training the models before the generated images collapse for Transfer GAN. As seen in Figure B, when the dataset is small, the Transfer GAN generates similar images, though the model can generate clearer images than our method. "Update all" just generate pixel-wise interpolation of training samples, which is apparent for flower dataset. This is discussed in the next section. Both methods generate images with better quality and diversity as the dataset sizes become large.

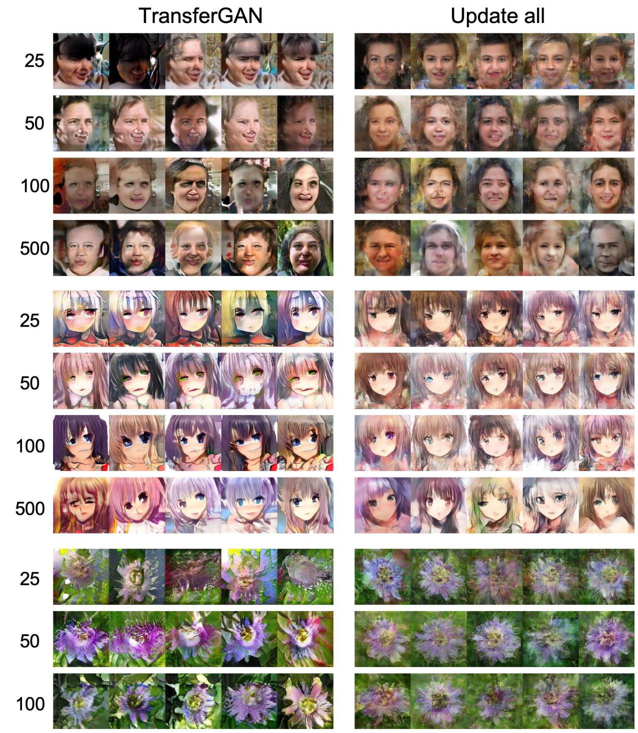

Figure B: Generated images from Transfer GAN and "Update all" trained with 25, 50, 100, 500 training images.

D. Comparison of interpolation results

In this section, we show the interpolation results for each dataset when the models are trained on 25 training samples. In Figure C, D, E, the top four rows show the interpolation between two randomly sampled images, and the bottom four rows show the interpolation between two generated images corresponding to two training samples.

The methods other than Transfer GAN, "Update all", and ours generate images with limited quality. Transfer GAN seems to generate more consistent images but the generated images are collapsed to a few modes according to the random generation results and evaluation scores. "Update all" just can conduct almost pixel-wise interpolation between two images. This is apparent for the hair change of human face and inconsistent shape of flower images. On the other hand, our method can perform more consistent interpolation between two images, although they are a little blurry.

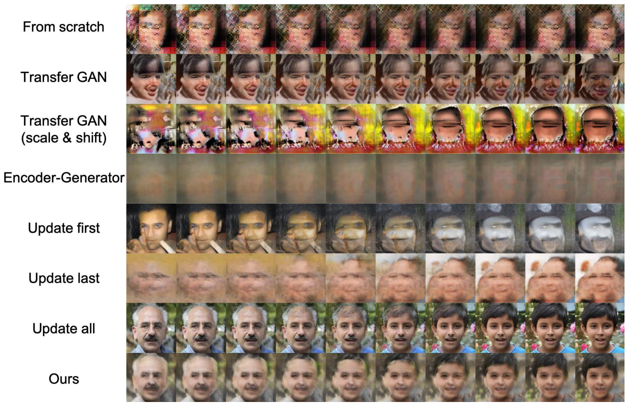

Figure C: Generated human face images from all methods trained with 25 training images.

Figure D: Generated anime face images from all methods trained with 25 training images.

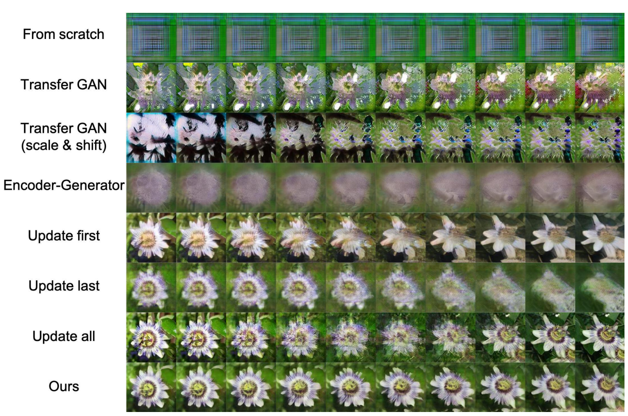

Figure E: Generated flower images from all methods trained with 25 training images.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In ICLR, 2019.

[2] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576, 2015.

[3] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In ICLR, 2018.

[4] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In ICLR, 2018.

[5] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[6] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring GANs: generating images from limited data. In ECCV, 2018.