Mode Seeking Generative Adversarial Networks for Diverse Image Synthesis Supplementary Material

1. Overview

In this supplementary material, we first present the implementation details of applying the proposed mode seeking regularization scheme on different baseline models. Second, we detail the procedure for using different quantitative evaluation metrics on various tasks. Third, we conduct an ablation study to analyze the impact of the proposed regularization term $\mathcal{L}_{\text{ms}}$ by varying the corresponding weight λ_{ms} for the training, and provide some results on the design choice of the distance metric. Then, the computational overheads are calculated to validate the efficiency of the proposed method. Finally, we demonstrate additional qualitative results to complement the paper.

2. Implementation Details

Table 1 summarizes the datasets and baseline models used on various tasks. For all of the baseline methods, we incorporate the original objective functions with the proposed regularization term. Note that we remain the original network architecture design and use the default setting of hyper-parameters for the training.

DCGAN. Since the images in the CIFAR-10 [4] dataset are of size 32×32, we modify the structure of the generator and discriminator in DCGAN [6], as shown in Table 2. We use the batch size of 32, learning rate of 0.0002 and Adam [3] optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ to train both the baseline and MSGAN network.

Pix2Pix. We adopt the generator and discriminator in BicycleGAN [12] to build the Pix2Pix [2] model. Same as BicycleGAN, we use a U-Net network [8] for the generator, and inject the latent codes $\mathbf{z}$ into every layer of the generator. The architecture of the discriminator is a two-scale PatchGAN network [2]. For the training, both Pix2Pix and MSGAN framework use the same hyper-parameters as the officially released version [1].

DRIT. DRIT [5] involves two stages of image-to-image translations in the training process. We only apply the mode seeking regularization term to generators in the first stage, which is modified on the officially released code [2].

StackGAN++. StackGAN++ [11] is a tree-like structure with multiple generators and discriminators. We use the output images from the last generator and input latent codes to calculate the mode seeking regularization term. The implementation is based on the officially released code [3].

3. Evaluation Details

We employ the official implementation of FID [4], NDB and JSD [5], and LPIPS [6]. For NDB and JSD, we use the K-means method on training samples to obtain the clusters. Then the generated samples are assigned to the nearest cluster to compute the bin proportions. As suggested by the author of [7], there are at least 10 training samples for each cluster. Therefore, we cluster the number of bins $K \approx N_{\text{train}}/20$ in all tasks, where N_{train} denotes the number of training samples for computing the clusters. We have verified that the performance is consistent within a large range of K. For evaluation, we randomly generate N images for a given conditional context on various tasks. We conduct five independent trials and report the mean and standard derivation based on the result of each trial. More evaluation details of one trial are presented as follows.

[1] https://github.com/junyanz/BicycleGAN/
[2] https://github.com/HsinYingLee/DRIT
[3] https://github.com/hanzhanggit/StackGAN-v2
[4] https://github.com/bioinf-jku/TTUR
[5] https://github.com/eitanrich/gans-n-gmms
[6] https://github.com/richzhang/PerceptualSimilarity

Conditioned on Class Label. We randomly generate $N = 5000$ images for each class label. We use all the training samples and the generated samples to compute FID. For NDB and JSD, we employ the training samples in each class to calculate $K = 250$ clusters.

Conditioned on Image. We randomly generate $N = 50$ images for each input image in the test set. For LPIPS, we randomly select 50 pairs of the 50 images of each context in the test set to compute LPIPS and average all the values for this trial. Then, we randomly choose 100 input images and their corresponding generated images to form 5000 generated samples. We use the 5000 generated samples and all samples in training set to compute FID. For NDB and JSD, we employ all the training samples for clustering and choose $K = 20$ bins for facades, and $K = 50$ bins for other datasets.

Conditioned on Text. We randomly select 200 sentences and generate $N = 10$ images for each sentence, which forms 2000 generated samples. Then, we randomly select $N_{\text{train}} = 2000$ samples for computing FID, and clustering them into $K = 100$ bins for NDB and JSD. For LPIPS, we randomly choose 10 pairs for each sentence and average the values of all the pairs for this trial.

4. Ablation Study on the Regularization Term

4.1. The Weighting Parameter λ_{ms}

To analyze the influence of the regularization term, we conduct an ablation study by varying the weighting parameter λ_{ms} on image-to-image translation task using the facades dataset. Figure 1 presents the qualitative and quantitative results. It can be observed that increasing λ_{ms} improves both the quality and diversity of the generated images. Nevertheless, as the weighting parameter λ_{ms} becomes larger than a threshold value (1.0), the training becomes unstable, which yields low quality, and even low diversity synthesized images. As a result, we empirically set the weighting parameter $\lambda_{\text{ms}} = 1.0$ for all experiments.

4.2. The Design Choice of the Distance Metric

We have explored other design choice of the distance metric. We conduct experiments using discriminator feature distance in our regularization term in a way similar to feature matching loss [10],

$$\mathcal{L}_{ms} = \frac{\frac{1}{L}\sum_{l=1}^{L} \|D^l(G(\mathbf{c}, \mathbf{z}_2)) - D^l(G(\mathbf{c}, \mathbf{z}_1))\|_1}{\|\mathbf{z}_2 - \mathbf{z}_1\|_1}, \tag{1}$$

where D^l denotes the l^{th} layer of the discriminator. We apply it to Pix2Pix on the facades dataset. Table. 3 shows that MSGAN using feature distance also obtains improvement over Pix2Pix. However, MSGAN using L_1 distance has higher diversity. Therefore, we employ MSGAN using L_1 distance for all experiments.

5. Computational Overheads

We compare MSGAN with Pix2Pix, BicycleGAN in terms of training time, memory consumption, and model parameters on an NVIDIA TITAN X GPU. Table. 4 shows that our method incurs marginal overheads. However, BicycleGAN requires longer time per iteration and larger memory with an additional encoder and another discriminator network.

6. Additional Results

We present more results of categorical generation, image-to-image translation, and text-to-image synthesis in Figure 3, Figure 4, Figure 5, Figure 6, Figure 7, and Figure 8, respectively.

Table 1: **Statistics of different generation tasks.** We summarize the number of training and testing images in each generation task. The baseline model used for each task is also provided.

Context	Class Label		Paired Images				Unpaired Images								Text	
Dataset	CIFAR-10 [4]		Facades [1]		Maps [2]		Yosemite [12]				Cat ⇌ Dog [5]				CUB-200-2011 [9]	
							Summer		Winter		Cat		Dog			
	train	test	train	test	train	test	train	test	train	test	train	test	train	test	train	test
Samples	50,000	10,000	400	206	1,096	1,098	1,069	309	770	238	771	100	1,264	100	8,855	2,933
Baseline	DCGAN [6]		Pix2Pix [2]				DRIT [5]								StackGAN++ [11]	

Table 2: **The architecture of the generator and discriminator of DCGAN**. We employ the following abbreviation: N= Number of filters, K= Kernel size, S= Stride size, P= Padding size. "Conv", "Dconv","BN" denote the convolutional layer, transposed convolutional layer and batch normalization, respectively.

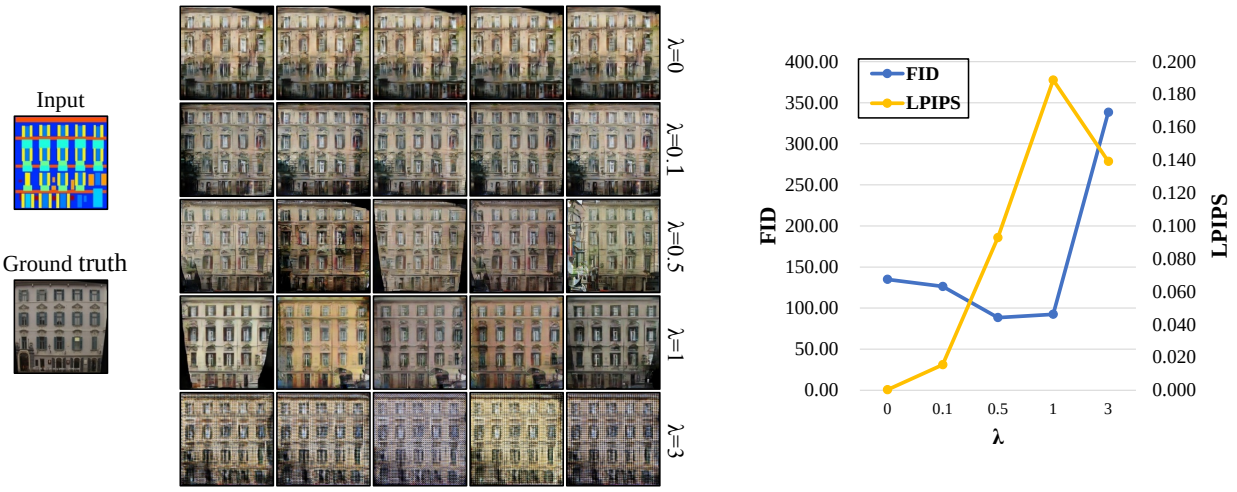
Layer	Generator	Discriminator
1	Dconv(N512-K4-S1-P0), BN, Relu	Conv(N128-K4-S2-P1), Leaky-Relu
2	Dconv(N256-K4-S2-P1), BN, Relu	Conv(N256-K4-S2-P1), BN, Leaky-Relu
3	Dconv(N128-K4-S2-P1), BN, Relu	Conv(N512-K4-S2-P1), BN, Leaky-Relu
4	Dconv(N3-K4-S2-P1), Tanh	Conv(N1-K4-S1-P0), Sigmoid

Table 3: **Quantitative results on the facades dataset.**

	Pix2Pix [2]	MSGAN-L_1	MSGAN-FD
FID ↓	139.19 ± 2.94	$\mathbf{92.84 \pm 1.00}$	100.16 ± 3.14
NDB↓	14.40 ± 1.82	12.40 ± 0.55	$\mathbf{11.80 \pm 1.48}$
JSD↓	0.074 ± 0.012	$\mathbf{0.038 \pm 0.004}$	0.072 ± 0.014
LPIPS↑	0.0003 ± 0.0000	$\mathbf{0.1894 \pm 0.0011}$	0.0565 ± 0.0003

Table 4: **Comparisons of computational overheads on the facades dataset.**

Model	Time (s)	Memory (MB)	Parameters (M)
Pix2Pix [2]	0.122	1738	58.254
MSGAN	0.122	1739	58.254
BicycleGAN [13]	0.192	2083	64.303



(a) Qualitative comparisons varying different λ

(b) FID and LPIPS values varying different λ

Figure 1: **Ablation study on the impact of λ_{ms}.** We show the (a) qualitative and (b) quantitative (FID and LPIPS) results. The study is conducted on image-to-image translation task with the facades dataset.

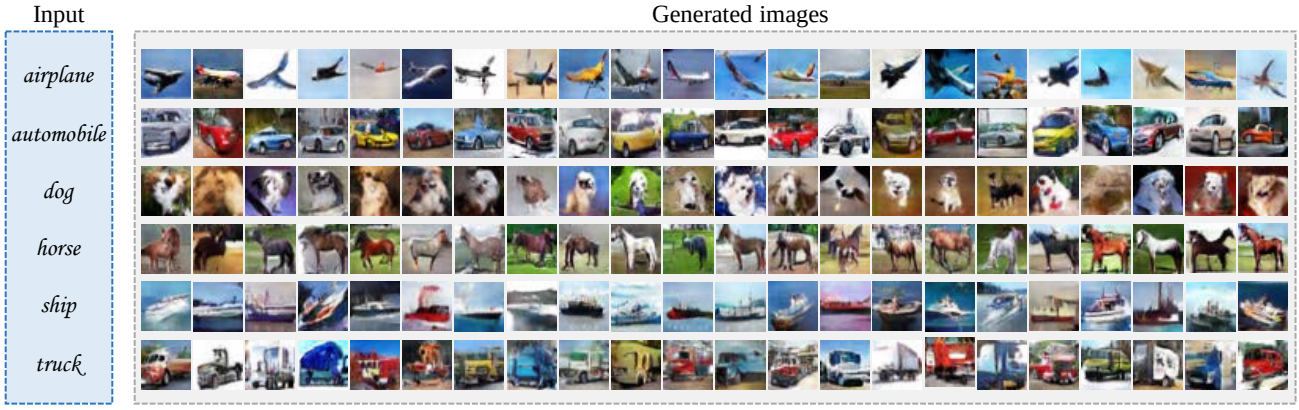


Figure 2: **More categorical generation results of CIFAR-10.** We show the results of DCGAN [6] with the proposed mode seeking regularization term on categorical generation task.

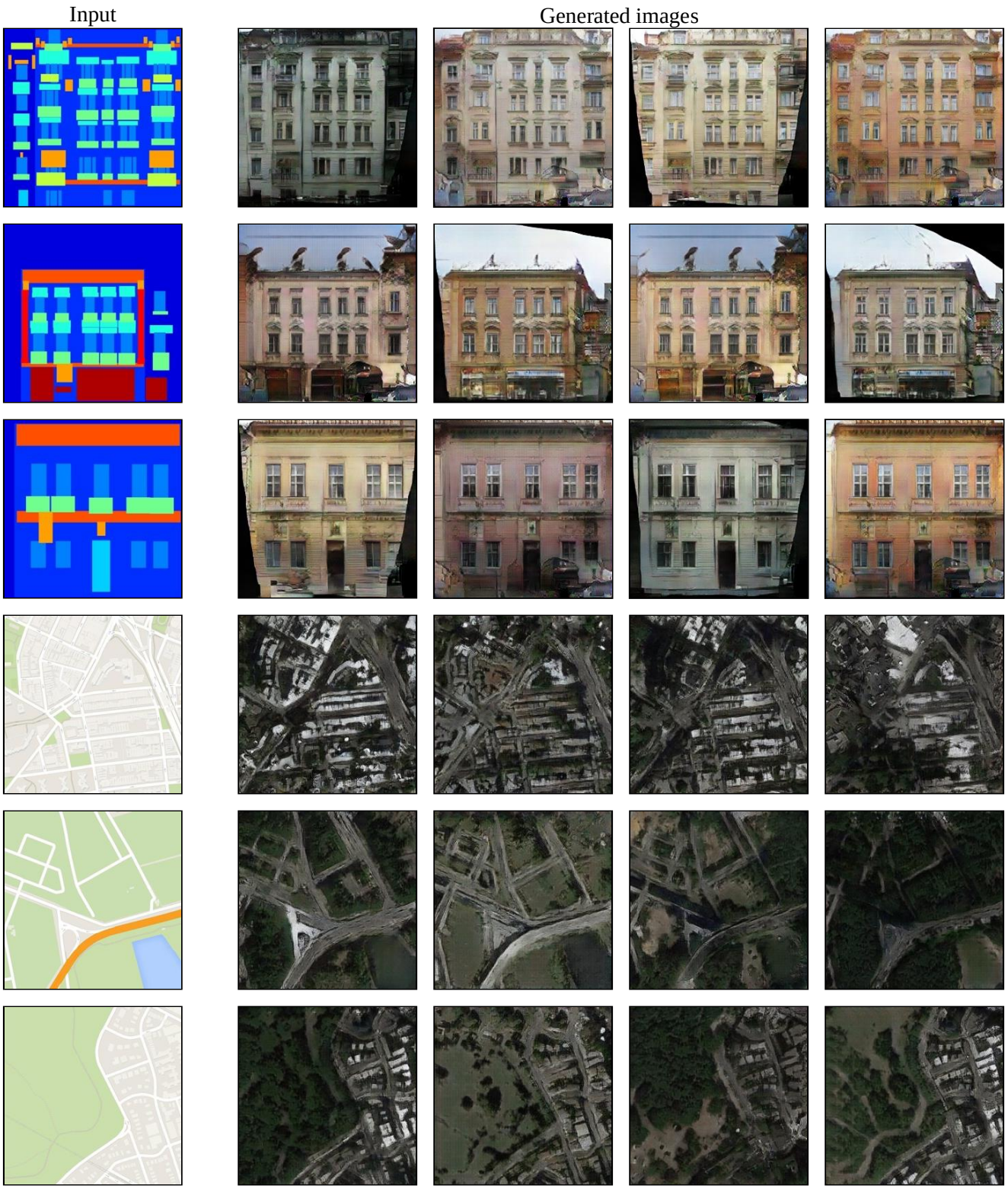


Figure 3: **More image-to-image translation results of facades and maps.** Top three rows: facades, bottom three rows: maps.

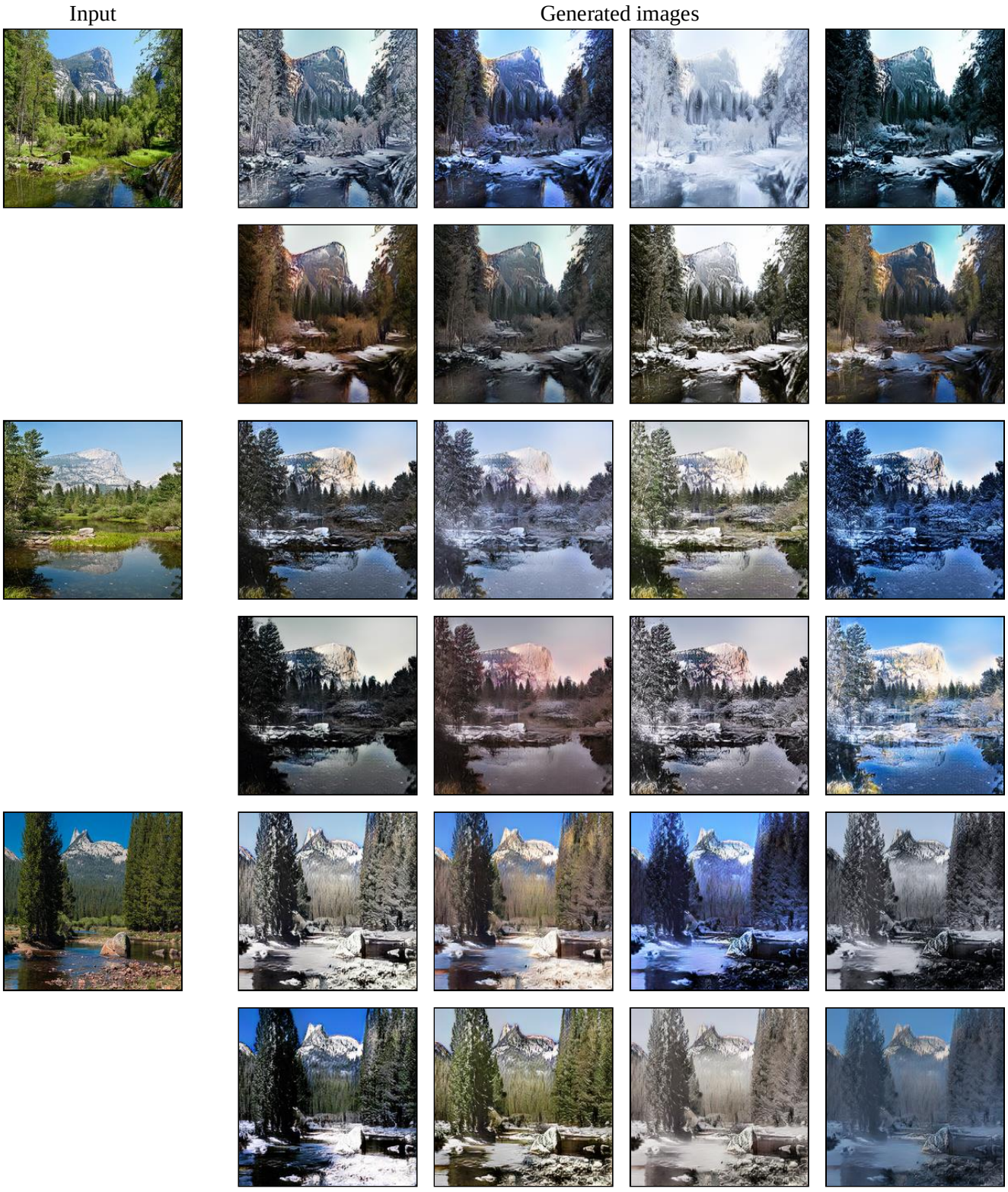


Figure 4: **More image-to-image translation results of Yosemite, Summer→Winter.**

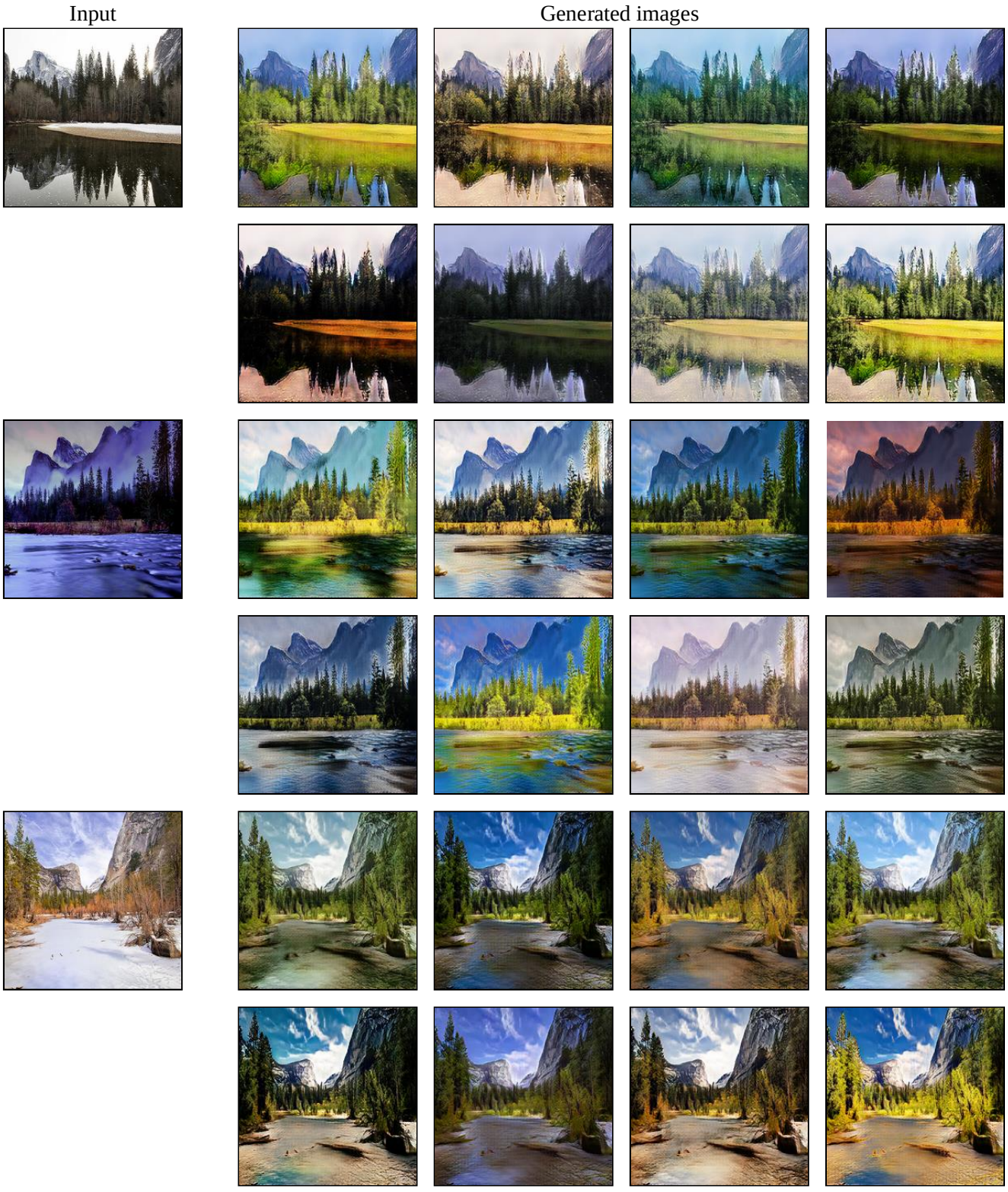


Figure 5: **More image-to-image translation results of Yosemite, Winter→Summer.**

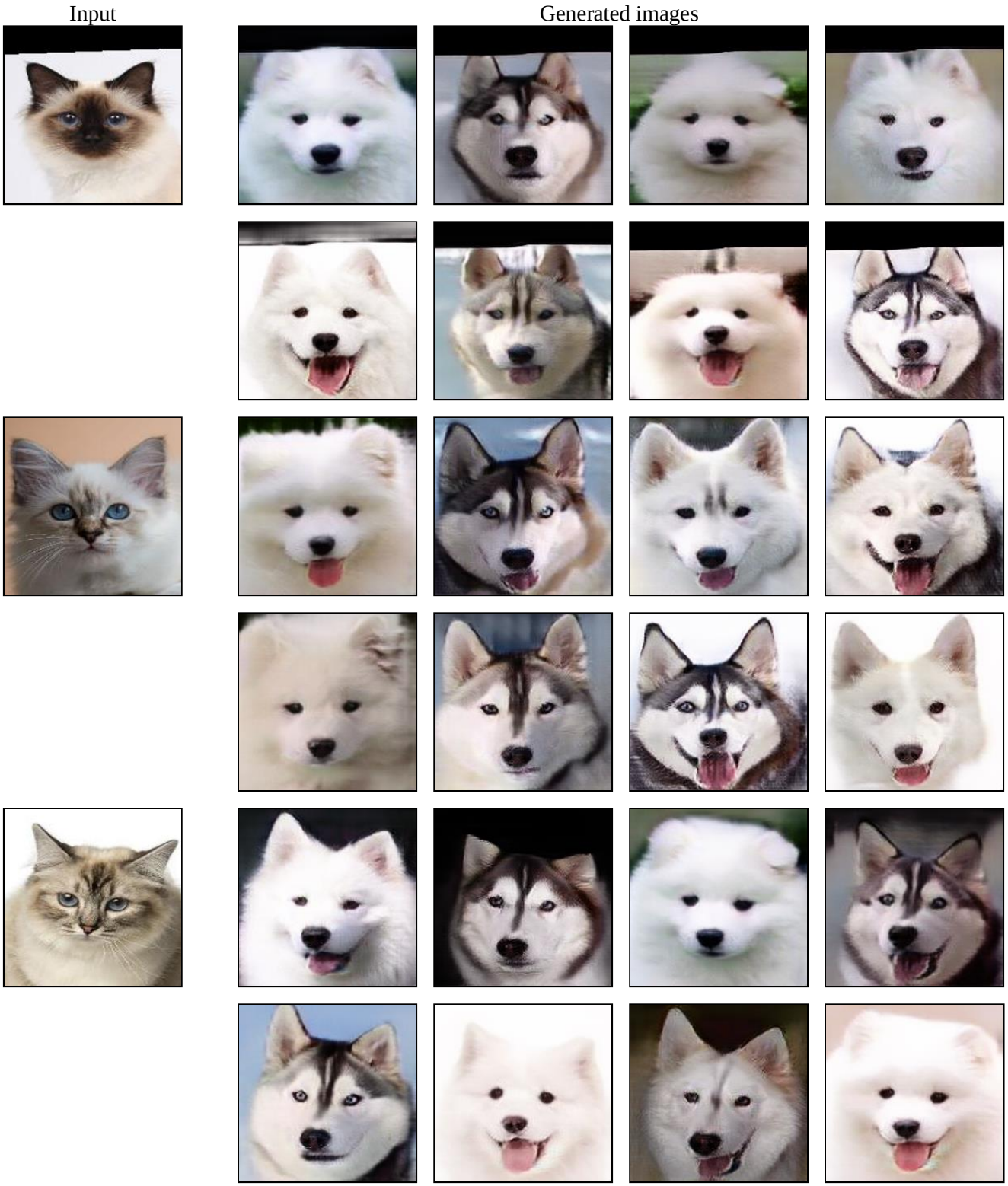


Figure 6: **More image-to-image translation results of Cat→Dog.**

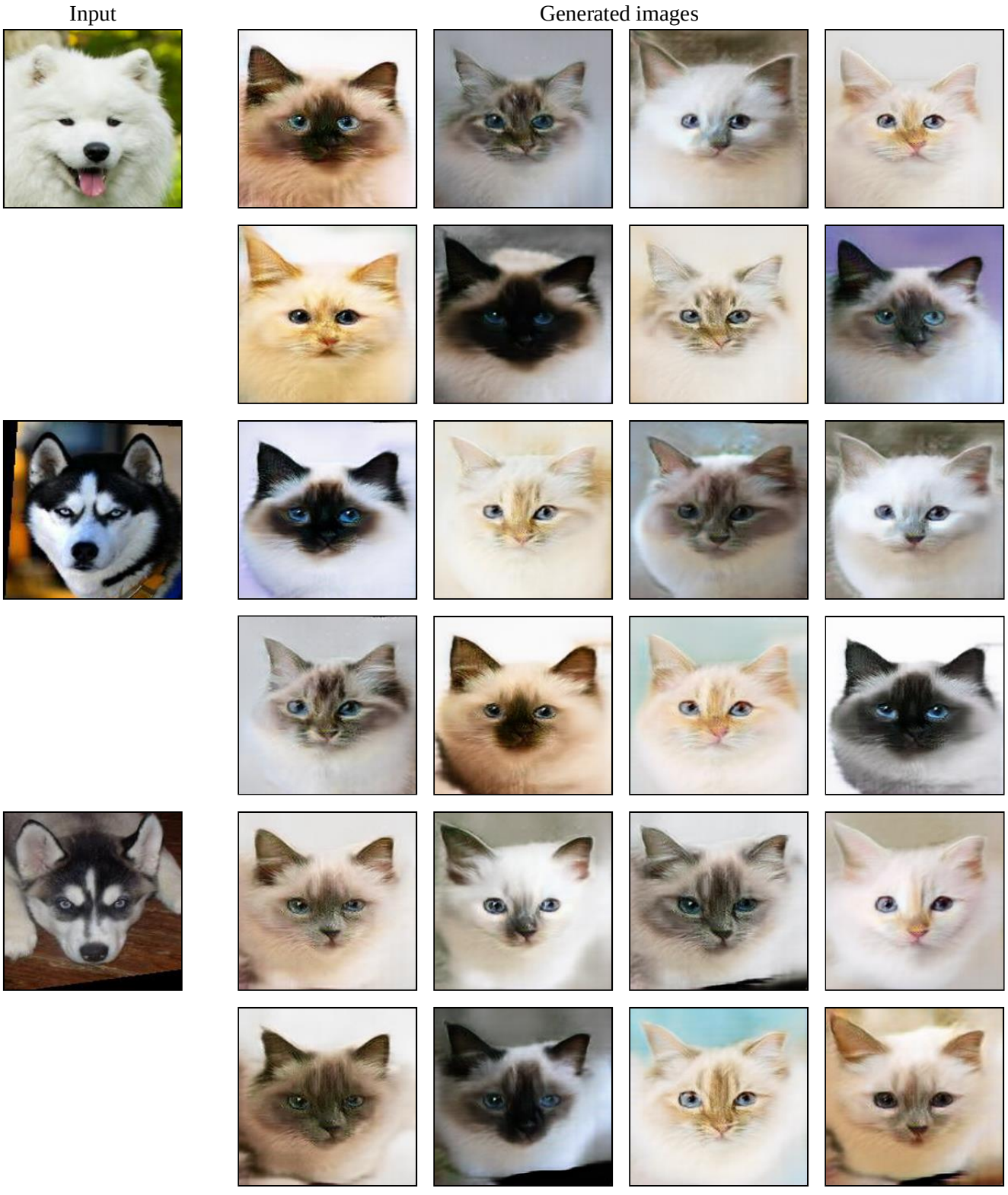


Figure 7: **More image-to-image translation results of Dog→Cat.**

Figure 8: **More text-to-image synthesis results of CUB-200-2011.**

References

[1] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 3

[2] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 1, 3

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

[4] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 1, 3

[5] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. K. Singh, and M.-H. Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018. 1, 3

[6] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 1, 3, 4

[7] E. Richardson and Y. Weiss. On GANs and GMMs. In *NIPS*, 2018. 1

[8] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. 2015. 1

[9] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 3

[10] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional GANs. In *CVPR*, 2018. 2

[11] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. StackGAN++: Realistic image synthesis with stacked generative adversarial networks. *TPAMI*, 2018. 1, 3

[12] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 1, 3

[13] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *NIPS*, 2017. 3