Supplementary Material: Semi-supervised Learning for Few-shot Image-to-Image Translation

1. SEMIT variants

We provide additional results for the experiments of SEMIT variants in Fig. 1.

2. Effect of OctConv layer

We provide additional results for the experiments on studying the effect of OctConv layer in Figs. 2 and 3.

3. Baselines

FUNIT [4] considers two training settings for the baselines, *fair* and *unfair*. Here, we only consider the unfair setting as it is more challenging. The unfair setting gives these baselines access (during training time) to the target images from the test data. In order to train CycleGAN and MUNIT, which are only able to perform I2I translation between two domains given during training, we define the source data as one domain and the target data as the other. Both StarGAN [1] and FUNIT [4] do not have this requirement as they approach multi-domain I2I translation.

4. Training settings

All models are implemented in PyTorch [6]. Our model architecture consists of six sub-networks: Pose encoder P_ϕ, Appearance encoder A_η, Generator G_Φ, Multilayer perceptron M_ω, Feature regulator F, and Discriminator D_ξ. The Pose encoder P_ϕ contains 3 OctConv layers and 6 blocks. Each OctConv layer uses 4×4 filters with stride 2, except for the first one which uses 7×7 with stride 1, and each block contains two OctConv layers with 3×3 filters and stride of 1. The Appearance encoder A_η consists of 3 OctConv layers, one average pooling layer and a 1×1 convolutional layer. M_ω consists of two fully connected layers with 256 and 4096 units (8-256-4096), and takes the output of the Appearance encoder as input. The Generator G_Φ comprises of ResBlock layers and two fractionally strided OctConv layers. The ResBlock consists of 6 residual blocks, as in the Pose encoder P_ϕ, but including AdaIN [2] layers. The AdaIN layers take the output of P_ϕ and the output of M_ω as input. The Feature regulator F, which takes the output of the Pose encoder as input, includes two average pooling layers with stride 2. For the D_ξ, we use one 7×7 convolutional layer with stride 1, and then 4 blocks for the Feature extractor $F_\Xi(\boldsymbol{x})$. Each block contains two ResBlocks, one average pooling layer with stride 2, and one 1×1 convolutional layer. The Feature extractor $F_\Xi(\boldsymbol{x})$ is followed by two parallel sub-networks, each of them containing one convolutional layer with 3×3 filters and stride 1. The OctConv operation contains high-frequency block ($H_{\tau'}$) and low-frequency block (L_τ). The former consists of two parallel branches: (a) the 3×3 convolutional layer with stride 1, and (b) one pooling layer and one convolutional layer which uses 3×3 with stride 1. The latter also contains two parallel branches: (a) the 3×3 convolutional layer with stride 1, and (b) one convolutional layer which uses 3×3 with stride 1 and one upsampling layer.

We randomly initialize the weights following a Gaussian distribution, and optimize the model using RMSProp with batch size 128, with a learning rate of 0.0001. Inspired by existing methods [4, 5], we employ the hinge loss variant of the GAN loss in our experiment. For the three hyper-parameters in Eq. 5 (τ_{cls}, τ_{cmp} and τ_{ent}), we use the same parameter values as in [3]. Two of the hyper-parameters in Eq. 8 (λ_a and λ_r) are set to the values used in FUNIT [4]. This demonstrates how our method is not particularly sensitive to hyper-parameter tuning and it works with the default values. The remaining parameters (λ_c and λ_e) are optimized with a line search 10^p with $p \in \{-5, -4, ..., 4, 5\}$. In all experiments, we use the following hyper-parameters: $\lambda_a = 1$, $\lambda_c = 0.1$, $\lambda_r = 0.1$, $\lambda_e = 0.1$. For the experiment on noise-tolerant pseudo-labeling, we use the following hyper-parameters: $\tau_{cls} = 1/C$, $\tau_{cmp} = 0.1/C$ and $\tau_{ent} = 0.8/C$, where C is the number of categories. We use 8 V100 GPUs in an NVIDIA DGX1 machine to perform all our experiments.

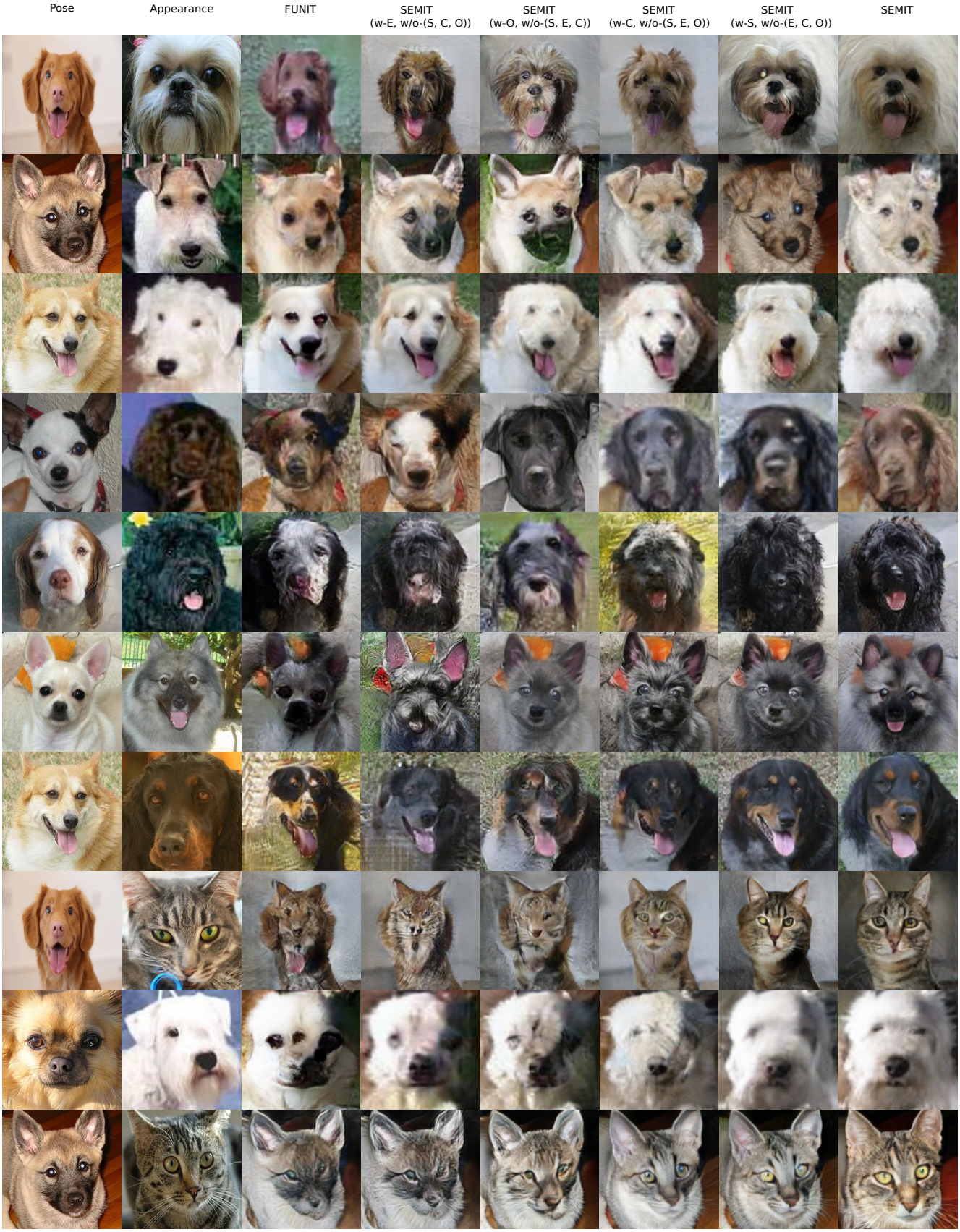

Figure 1. Comparison between FUNIT [4] and variants of our proposed method. For example, *SEMIT(w-E, w/o-(S, C, O))* indicates the model trained with only entropy regulation.

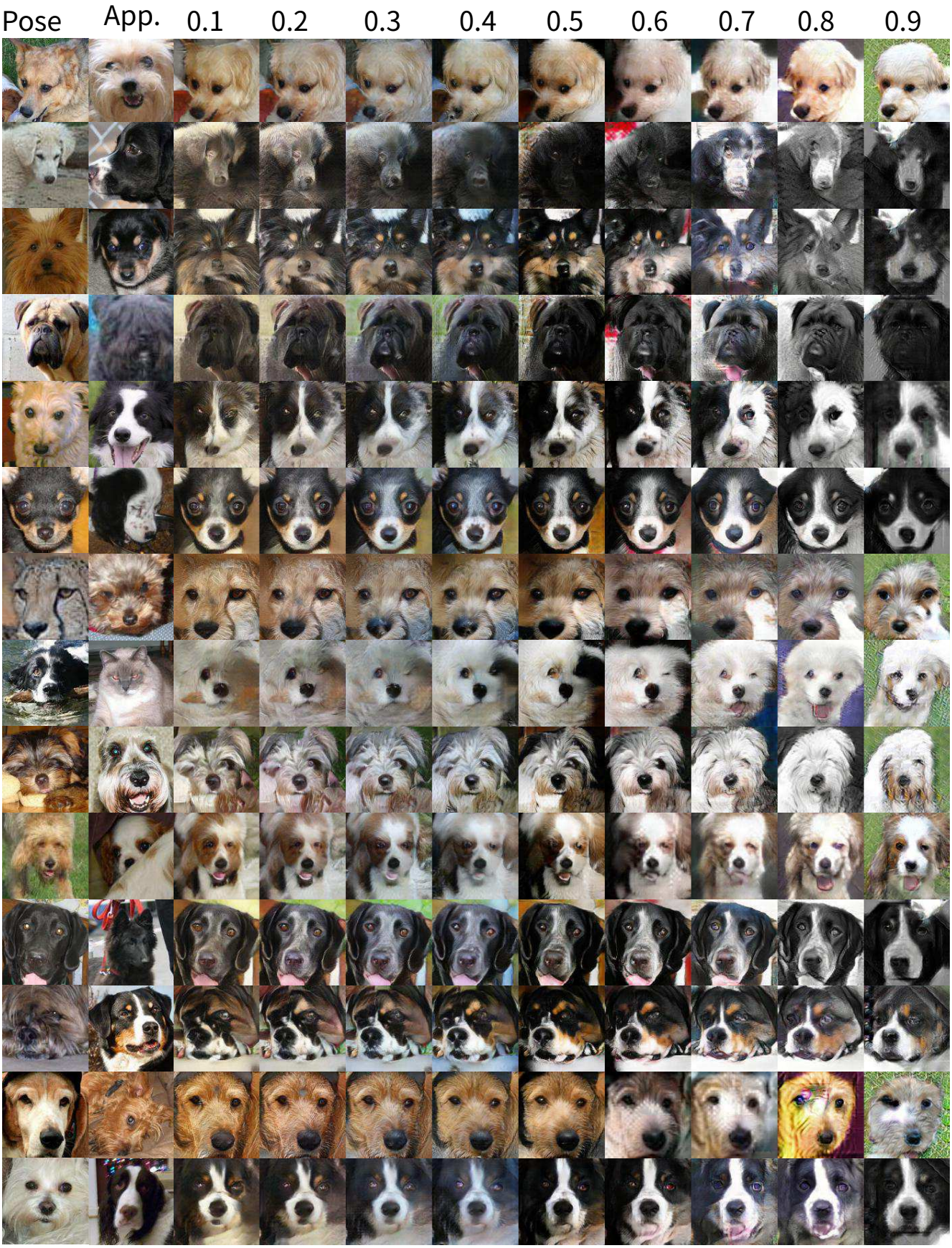

Figure 2. Qualitative results for several ratios of high/low frequency channels in OctConv. Results correspond to one-shot I2I translation on Animals-69 with 90% labeled data.

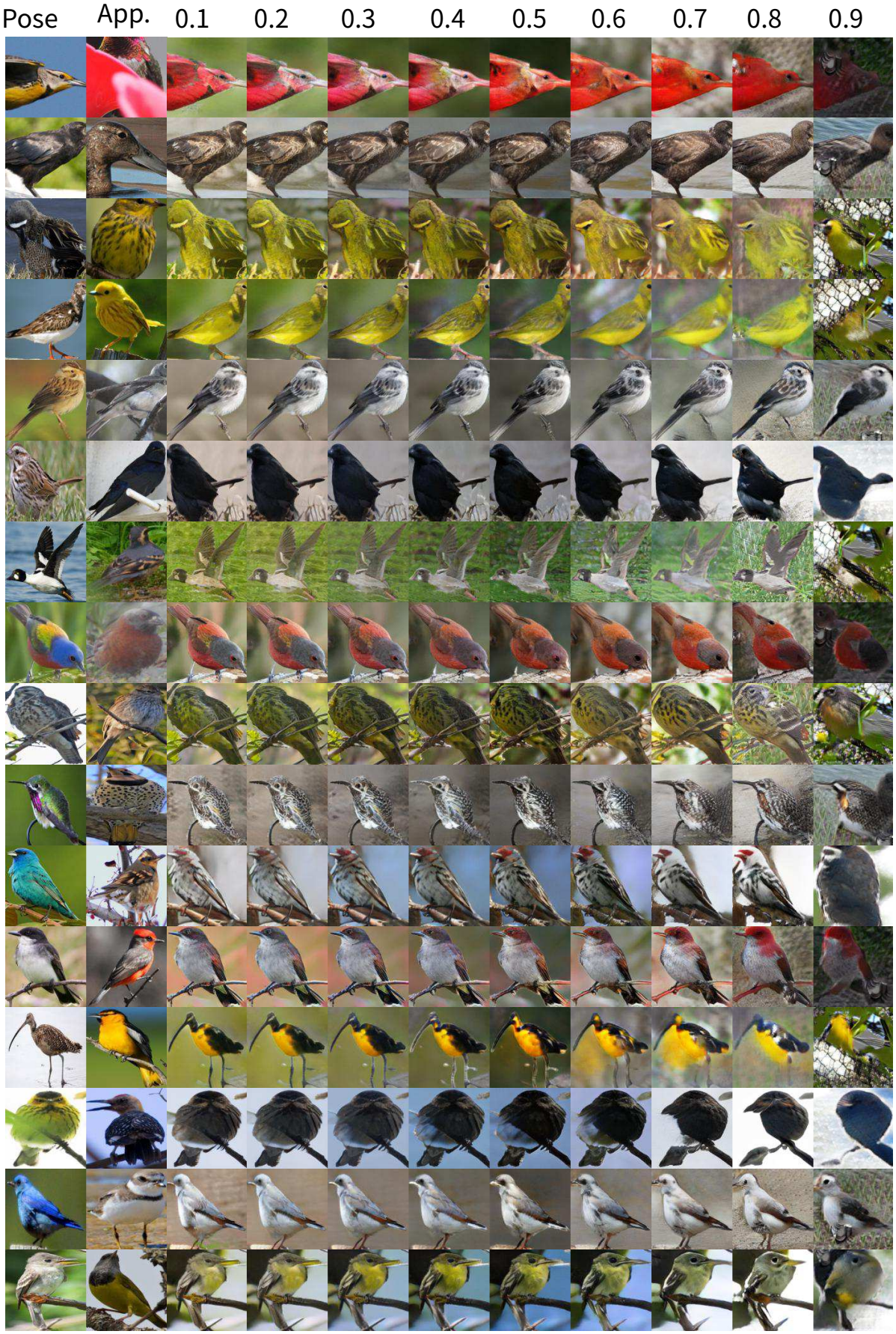

Figure 3. Qualitative results for several ratios of high/low frequency channels in OctConv. Results correspond to one-shot I2I translation on Birds with 90% labeled data.

	Setting	Top1-all	Top5-all	Top1-test	Top5-test	IS-all	IS-test	mFID
100%	SEMIT-1	29.42	65.51	62.47	90.29	24.48	13.87	75.87
	SEMIT-5	35.48	78.96	71.23	94.86	25.63	15.68	68.32
	SEMIT-20	45.70	88.5	74.86	**99.51**	26.23	16.31	49.84
	SEMIT-100	**45.93**	**89.6**	**76.01**	99.30	**28.54**	**17.26**	**47.98**
20%	SEMIT-1	26.71	69.48	65.48	85.49	23.52	12.63	92.21
	SEMIT-5	39.56	78.34	71.81	96.25	24.01	14.17	69.28
	SEMIT-20	44.25	85.60	73.80	98.62	24.67	15.04	65.21
	SEMIT-100	**45.97**	**87.58**	**74.59**	**98.96**	**26.67**	**16.07**	**64.52**
10%	SEMIT-1	16.25	51.55	39.71	81.47	22.58	8.61	99.42
	SEMIT-5	29.40	76.14	62.72	92.13	22.98	13.24	78.46
	SEMIT-20	39.02	82.90	69.70	95.40	23.43	14.07	69.40
	SEMIT-100	**39.91**	**84.06**	**70.58**	**95.99**	**24.04**	**14.89**	**67.78**

Table 1. Performance comparison with baselines on Animals [4].

	Setting	Top1-all	Top5-all	Top1-test	Top5-test	IS-all	IS-test	mFID
100%	SEMIT-1	15.64	42.85	43.7.62	72.41	69.63	20.12	105.82
	SEMIT-5	23.57	55.96	49.42	80.41	78.42	24.98	90.48
	SEMIT-20	**28.15**	62.41	54.62	83.32	82.64	27.51	83.56
	SEMIT-100	27.08	**64.7**	**55.31**	**83.80**	**83.47**	**27.94**	**81.03**
20%	SEMIT-1	13.58	48.16	43.97	64.27	59.29	16.48	109.84
	SEMIT-5	19.23	53.25	50.34	73.16	67.84	22.27	98.38
	SEMIT-20	21.49	57.55	52.34	76.41	72.31	**23.44**	95.41
	SEMIT-100	**21.62**	**57.79**	**53.36**	**76.97**	**74.26**	23.02	**93.81**
10%	SEMIT-1	11.21	37.14	35.14	59.41	48.48	12.57	128.4
	SEMIT-5	13.54	43.63	40.24	68.75	59.84	17.58	119.4
	SEMIT-20	15.41	48.36	42.51	71.49	**65.42**	19.87	109.8
	SEMIT-100	**16.87**	**50.68**	**42.65**	**73.64**	64.85	**20.45**	**108.1**

Table 2. Performance comparison with baselines on Birds [7].

	Setting	Top1-all	Top5-all	Top1-test	Top5-test	IS-all	IS-test	mFID
10%	FUNIT-1	10.21	28.41	27.42	49.54	17.24	4.05	156.8
	FUNIT-5	13.04	35.62	31.21	61.70	19.12	4.87	138.8
	FUNIT-20	14.84	39.64	37.52	65.84	19.64	5.53	127.8
	Ours(SNG)-1	16.25	51.55	39.71	81.47	22.58	8.61	99.42
	Ours(SNG)-5	29.40	76.14	62.72	92.13	22.98	13.24	78.46
	Ours(SNG)-20	39.02	82.90	69.70	95.40	23.43	14.07	69.40
	Ours(JNT)-1	18.42	53.86	41.23	85.59	22.47	10.61	88.62
	Ours(JNT)-5	31.52	79.25	63.81	94.54	23.05	14.83	69.57
	Ours(JNT)-20	**51.26**	**83.51**	**73.86**	**96.74**	**24.85**	**15.64**	**62.31**

Table 3. Performance comparison with baselines on Animals++.

	Setting	Top1-all	Top5-all	Top1-test	Top5-test	IS-all	IS-test	mFID
10%	FUNIT-1	6.04	19.34	12.51	38.84	32.62	7.47	203.3
	FUNIT-5	8.82	22.52	19.85	42.53	38.59	9.53	175.7
	FUNIT-20	10.98	26.41	22.48	48.36	41.37	13.85	154.9
	Ours(SNG)-1	11.21	37.14	35.14	59.41	48.48	12.57	128.40
	Ours(SNG)-5	13.54	43.63	40.24	68.75	59.84	17.58	119.44
	Ours(SNG)-20	15.41	48.36	42.51	71.49	65.42	19.87	109.81
	Ours(JNT)-1	14.69	42.15	36.72	65.19	67.48	18.12	108.97
	Ours(JNT)-5	21.92	55.86	48.17	76.33	73.97	23.84	96.23
	Ours(JNT)-20	**26.23**	**61.34**	**52.68**	**79.97**	**78.31**	**25.49**	**95.41**

Table 4. Performance comparison with baselines on Birds++.

5. Quantitative results for many-shot

We also provide results for *many-shot* on a single dataset in Tabs. 1, 2. Given more appearance images per class, our method obtains a higher performance. But the gap between 100-shot and 20-shot is small, clearly indicating that our method already obtains good results while using only few appearance images.

6. Qualitative results for models trained on a single dataset

We provide additional results for models trained on a single dataset in Figs. 4, 5, 6, 7.

7. Qualitative results for models trained on multiple datasets

We provide additional results for models trained on multiple datasets in Fig. 8.

8. Quantitative results for models trained on multiple datasets

We provide additional quantitative results for models trained on multiple datasets in Tabs. 3 and 4.

References

[1] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[2] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision*, pages 172–189, 2018.

[3] Yi Kun and Wu Jianxin. Probabilistic End-to-end Noise Correction for Learning with Noisy Labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[4] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10551–10560, 2019.

[5] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *International Conference on Learning Representations*, 2018.

[6] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[7] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604, 2015.

Figure 4. Qualitative comparison on the Animals datasets.

Figure 5. Qualitative comparison on the Birds datasets.

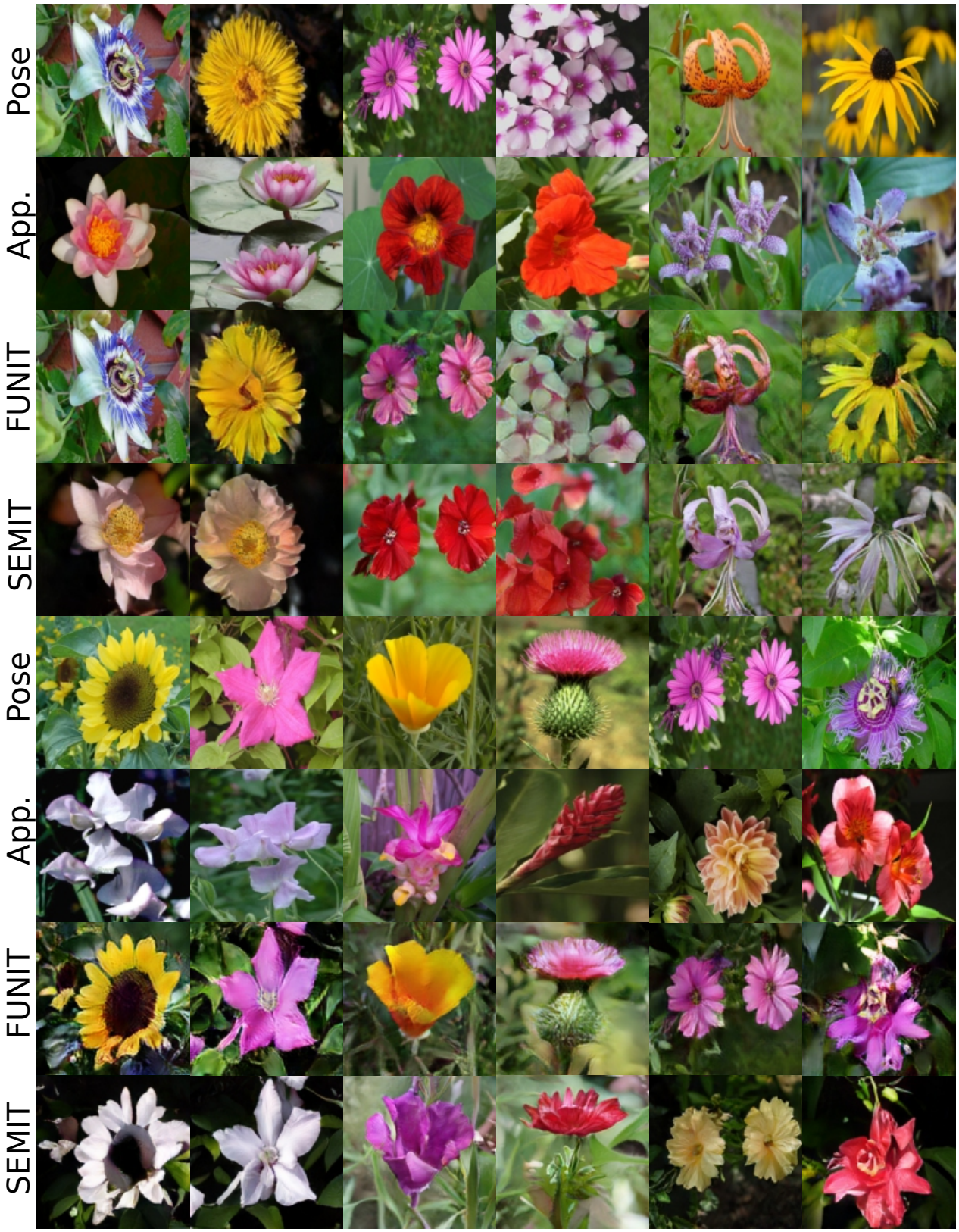

Figure 6. Qualitative comparison on the Flowers datasets.

Figure 7. Qualitative comparison on the Foods datasets.

Figure 8. Results of our method on a single dataset (SNG) and joint datasets (JNT).