A Style-aware Discriminator for Controllable Image Translation - Appendix

Kunhee Kim Sanghun Park Eunyeong Jeon Taehun Kim Daijin Kim
Pohang University of Science and Technology (POSTECH)
{kunkim,sanghunpark,eyjeon,taehoon1018,dkim}@postech.ac.kr

A. Implementation

A.1. Architecture

The overall architecture of our method follows StarGAN v2 [2]. We normalized the output content code in each pixel to the unit length following Park *et al*. [9]. When using the StyleGAN2-based generator, we replaced the instance normalization of the content encoder with pixel normalization [4]. We did not use an equalized learning rate [6].

The style-aware discriminator consists of $M = 0.25 * \log_2(\text{resolution})$ residual blocks followed by an average pooling. The style head and the discrimination head are two-layer MLPs. We used the same discriminator for the StyleGAN2-based and AdaIN-based models. We set dimension of prototypes to 256. We set K to 32 for AFHQ, 64 for CelebA-HQ, and 128 for LSUN churches and FFHQ.

A.2. Augmentation

Geometric transform We used the `RandomRation` and `RandomScaleAdjustment` augmentations. We applied reflection padding to avoid empty areas in the image before applying the geometric transform. We chose the rotation angle to be between -30 and 30 and the scale parameter between 0.8 and 1.2. Each transform was applied with a probability of 0.8.

Cutout The style of the human face domain is integral to characteristics other than color and texture, including gender, expression, and accessories. We can read such information (*i.e*., the style) from an image even when part of a human face is occluded. Accordingly, we employed cutout augmentation. In practice, we used the `RandomErasing` method from the `torchvision` library with the following probability and scale parameters: `p=0.8` and `scale=(0.1, 0.33)`.

Color distortion We observed that when the variation of the dataset is significant (*e.g*., FFHQ) or when the batch size is small, it was not possible to manipulate short hair into long hair. In that case, we employed weak color jittering. More specifically, we applied the `ColorJitter` method with the following parameters with a probability of 0.8: `brightness=0.2`, `contrast=0.2`, `saturation=0.2`, `hue=0.01`. Note that, we applied this augmentation only with the CelebA-HQ dataset using AdaIN and the FFHQ experiments.

A.3. Style code sampling

We sampled the style code from the dataset $\mathcal{X}$ with a probability p. Otherwise, we sampled from the prototypes. When sampling from a dataset, we used a randomly shuffled minibatch $\mathbf{x}'$ to create a style code $\tilde{\mathbf{z}}_s = f_s(\mathbf{x}')$. In the case of sampling from the prototypes, we used the following pseudocode. In practice, we set p to 0.8 except in the case of for longer training (25 M), where we used 0.5.

```
# C: prototypes (K x D)
# N: batch size
# K: number of prototypes
# D: prototype dimension

@torch.no_grad()
def sample_from_prototypes(C, N, eps=0.01):
    K, D = C.shape

    samples = C[torch.randint(0, K, (N,))]
    if torch.rand(1) < 0.5:  # perturbation
        eps = eps * torch.randn_like(samples)
        samples = samples + eps
    else:  # interpolation
        targets = C[torch.randint(0, K, (N,))]
        t = torch.rand((N, 1))
        samples = torch.lerp(samples, targets, t)
    return F.normalize(samples, p=2, dim=1)
```

A.4. Training details

In every iterations, we sampled a minibatch $\mathbf{x}$ of N images from the dataset. To calculate the *swapped prediction loss*, we created two different views $\mathbf{x}_1 = \mathcal{T}_1(\mathbf{x}), \mathbf{x}_2 = \mathcal{T}_2(\mathbf{x})$, where $\mathcal{T}$ is an augmentation. We reused the $\mathbf{x}_1$ as the input of the generator. We obtained style codes by sampling the prototype with probability p or encoding reference images $\mathbf{x}' = \texttt{shuffle}(\mathbf{x}_1)$ with probability $(1-p)$. In practice, we usually set p as 0.8, but 0.5 when training is long enough (longer than 5 M). When sampling from the prototype, the first two of Eq. 2 was selected uniformly. The adversarial loss for updating the discriminator D was calculated for $G(\mathbf{x}_1, \mathbf{s})$, and the adversarial loss for updating

Method	FID	
	Churches	**FFHQ** 256^2
Ours (latent)	**9.0**	5.2
Ours (reference)	12.2	**5.1**
*SwapAE [9]	49.6	-
StyleGAN2 [6]	4.1	3.7

Table 6. Quantitative comparison using the unlabeled datasets. An asterick (*) indicates that we used the pre-tranined networks provided by the authors. Note that we calculated StyleGAN2 results using randomly sampled images, not manipulated images (*i.e.* style mixing).

Method	FID_{lerp}	
	AFHQ	**CelebA-HQ**
Ours	**11.2**	**25.4**
Ours-AdaIN	***14.0***	***31.0***
Liu *et al.* [8]	30.0	35.8
StarGAN v2 [2]	32.2	76.8

Table 7. Quantitative comparison of the style interpolation.

the generator G was calculated for $G(\mathbf{x}_1, \mathbf{x})$ and the reconstructed image.

We applied the lazy R1 regularization following [6]. To stabilize the SwAV training, we adopted training details from the original paper [1]. In more detail, we fixed the prototype for the first 500 iterations and used the queue after the 20,000th iteration if $K < N$. We linearly ramped up learning rate for the first 3000 iterations.

We initialized all of the networks using Kaiming initialization [3]. Following Choi *et al.* [2], we used ADAM [7] with a learning rate of 0.0001, $\beta_1 = 0.0$ and $\beta_2 = 0.99$. We scaled the learning rate of the mapping network by 0.01, similar to previous studies [2, 5]. By default, we used a batch size of 16 for the AdaIN-based model and 32 for the StyleGAN2-based model. We used a larger batch size (64) and longer training (25 M) for the FFHQ and LSUN churches datasets. We observed that the performance improves as the batch size and the number of training images increase.

B. Additional results

B.1. Quantitative results for the unlabeled datasets

We measured the quality of the latent-guided and reference-guided synthesis on the unlabeled datasets in Table 6. The proposed method significantly outperforms the Swapping Autoencoder [9] on the LSUN churches validation set. For reference, we also report the results of unconditionally generated StyleGAN2 images. Even though the proposed method is inferior to unconditional GANs (*i.e.*, StyleGAN2 [6]), note that unconditional GANs are unsuitable for image editing [9].

Figure 7. Qualitative comparison of the style interpolation. We sampled three images (one source and two references) from the dataset and synthesized images using the style code interpolated between the two style codes obtained from the two reference images.

B.2. Quality of the style interpolation

To evaluate the quantitative results of the style interpolation, we calculated FID between the training set and images synthesized using interpolated styles (FID_{lerp}). We sampled images from two different domains and generated ten style codes by interpolating their corresponding style code. Then, we synthesized ten images using those style codes (we used the first sample as a source image). We created 30,000 fake images for the AFHQ and a total of 20,000 fake images for CelebA-HQ. As shown in Table 7, the proposed method outperforms the supervised approaches [2, 8] in terms of FID. Fig. 7 shows the qualitative comparison between the proposed model and baselines. The proposed approach was the only model that produced smooth interpolation results while maintaining the content such as back-

Figure 8. Qualitative results for the `separated` method.

K	32	64	128	256	512	1024
k-NN↑	99.1	99.1	98.8	98.5	96.3	95.6
mFID↓	14.7	15.8	28.4	26.6	34.6	42.1

Table 8. Effect of the number of prototypes. Note that mFID of supervised method (StarGAN v2) [2] is 24.1.

grounds.

B.3. Additional qualitative results

Here, we include qualitative results for various datasets. Fig. 9 shows the results of the model trained at 512×512 resolution on the AFHQ v2 dataset. Fig. 10 and 11 show the reference-guided image synthesis results on unlabeled datasets (FFHQ and LSUN churches). Fig. 12 shows the reference-guided image synthesis results for the Oxford-102 dataset. Finally, we visualize the all prototypes learned with the AFHQ and CelebA-HQ datasets in Fig. 13.

C. Additional analyses

C.1. Effect of the style-aware discriminator

The low k-NN metric of the `separated` method implies that the style space is not highly correlated with the species. This is further supported by the qualitative results. As shown in Fig. 8, the `separated` method learns to translate the tone of the image rather than desired style (*i.e.*, the species), which explains the very high mFID[1].

C.2. Ablation based on the number of prototypes

In Table 8, we evaluate the effect of the number of prototypes (K) on the proposed method. We trained the AdaIN-based model with varying K using the AFHQ dataset. We observed that the appropriate number of prototypes was critical to the synthesis quality. However, even when the value of K was large, the mFID value did not deviate from a certain range. We did not conduct experiments to determine the optimal value of K for the other datasets; instead, we set the value of k based on the number of images in the dataset.

[1] In the AFHQ dataset, the models that cannot change species result in high mFID, since the FID between different species can be rather large. For example, the FID between a real cat and real dog is 170.4.

References

[1] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 2

[2] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, 2020. 1, 2, 3

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 2

[4] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 1

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 2

[6] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 1, 2

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 2

[8] Yahui Liu, Enver Sangineto, Yajing Chen, Linchao Bao, Haoxian Zhang, Nicu Sebe, Bruno Lepri, Wei Wang, and Marco De Nadai. Smoothing the disentangled latent style space for unsupervised image-to-image translation. In *CVPR*, 2021. 2

[9] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. In *NeurIPS*, 2020. 1, 2

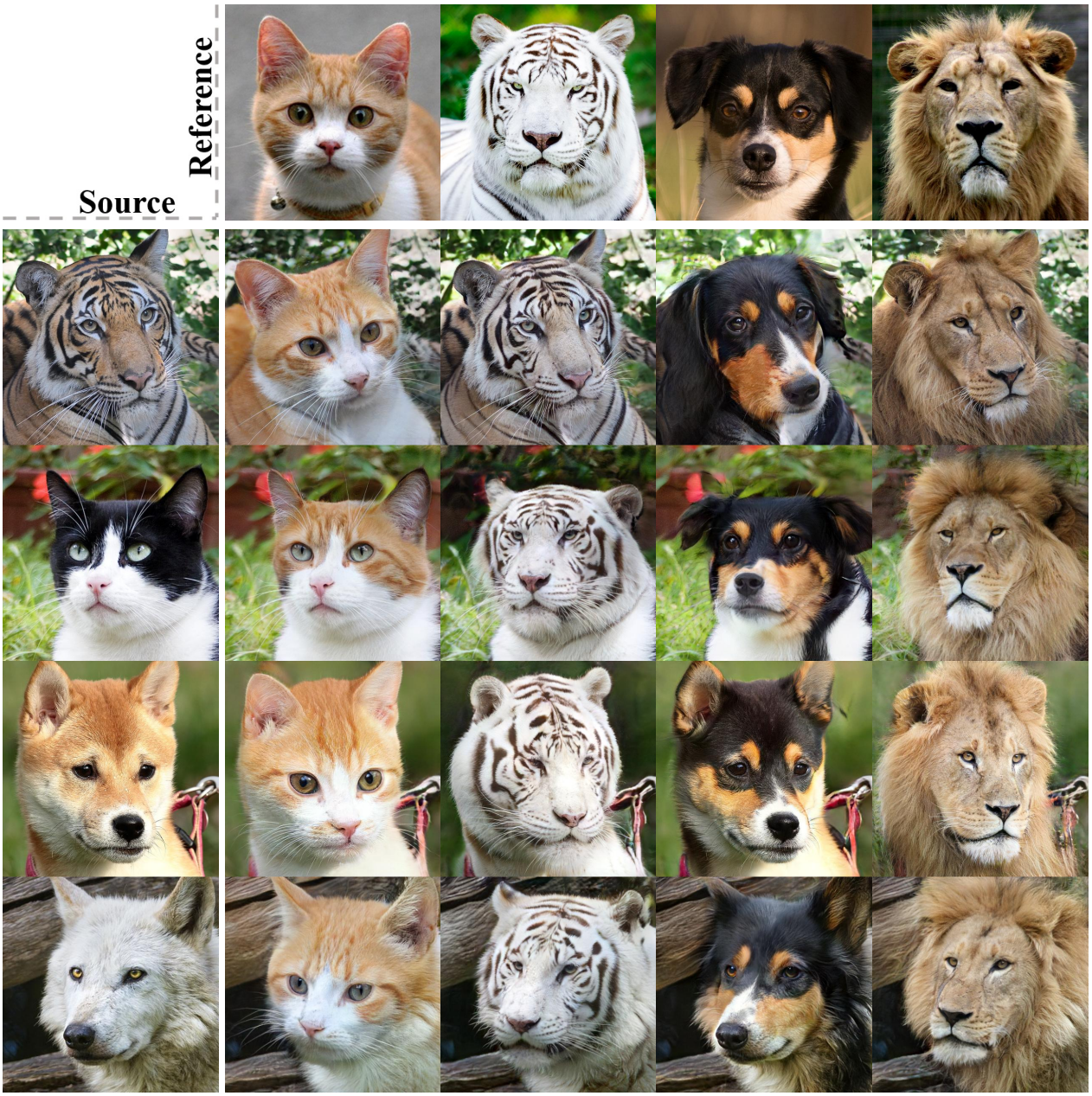

Figure 9. Reference-guided synthesis results on the AFHQ v2 dataset. The model was trained and tested at 512×512 resolution.

Figure 10. Reference-guided synthesis results on the FFHQ dataset.

Figure 11. Reference-guided synthesis results on the LSUN churches dataset. The model was trained at 256×256 resolution and tested at 256 resolution on the shorter side.

Figure 12. Reference-guided synthesis results on the Oxford-102 dataset. The model was trained and tested at 256×256 resolution.

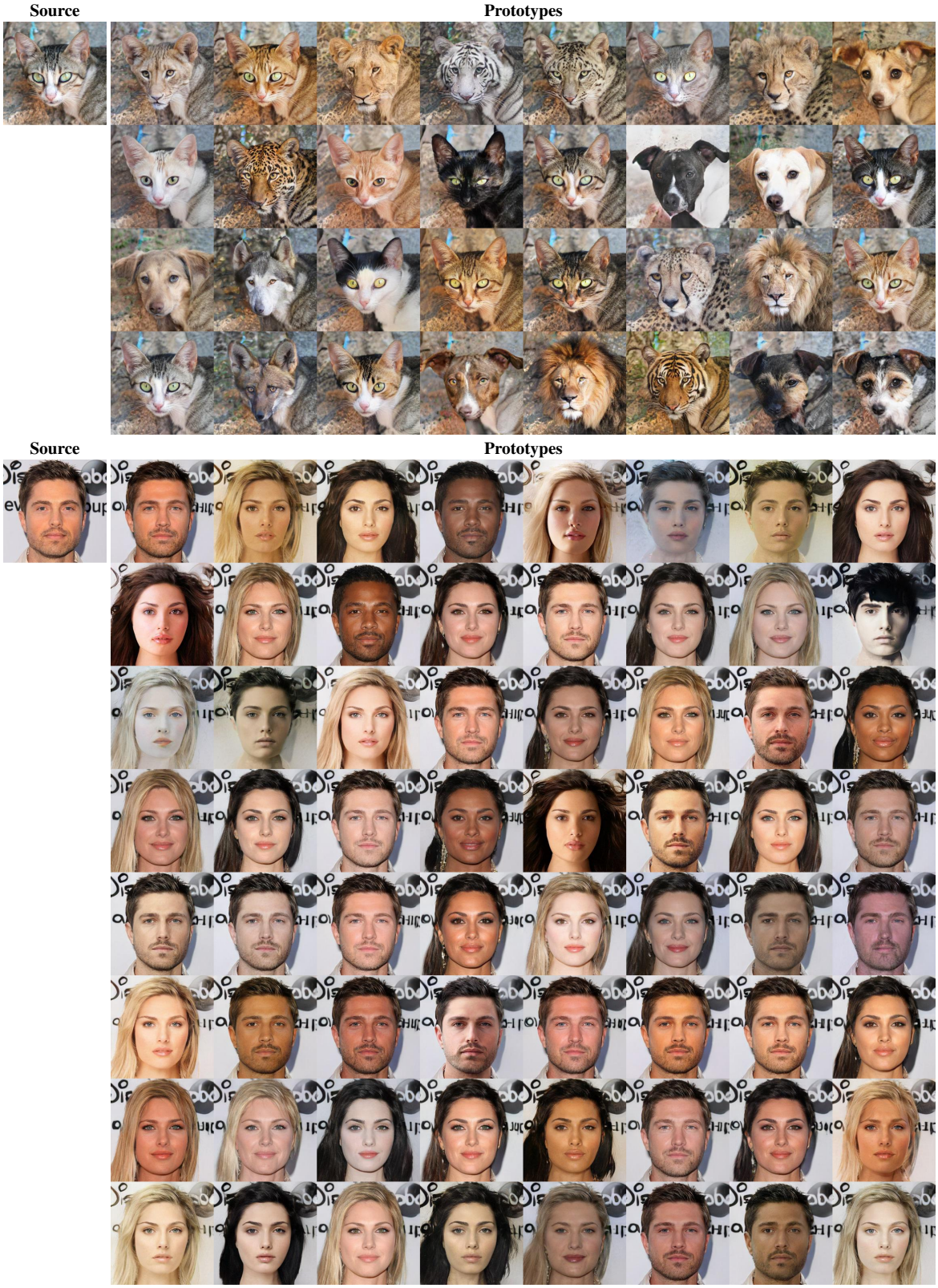

Figure 13. Visualization of all prototypes. (Top) 32 prototypes learned with the AFHQ dataset. (Bottom) 64 prototypes learned with the CelebA-HQ dataset.