Mind the Gap in Distilling StyleGANs

Guodong Xu[1], Yuenan Hou[2], Ziwei Liu[3], and Chen Change Loy[3]

[1] The Chinese University of Hong Kong
[2] Shanghai AI Laboratory
[3] S-Lab, Nanyang Technological University
xg018@ie.cuhk.edu.hk, houyuenan@pjlab.org.cn,
{ziwei.liu, ccloy}@ntu.edu.sg

1 Implementation Details

Training Hyperparameters. For the mapping network mimicking of the first stage, we use Adam as the optimizer with a initial learning rate of 0.05. We train for 50k steps and the batch size is set as 4096. For the normal GAN training of the second stage, we use Adam optimizer with a initial learning rate of 0.002 and 450k iterations. For the α that controls the offset along latent direction, we sample it from a Gaussian distribution $\mathcal{N}(0, 5)$. We set λ_{GAN}, λ_{rgb}, λ_{lpips} and λ_{LD} to be 1, 3, 3 and 30, respectively. The features that are used to compute LD loss come from the outputs of 64/128/256 resolution blocks.
Evaluation Metrics. Fréchet Inception Distance (FID) is a commonly used metric to evaluate the realism of generated images. The generated images and real images are fed into a inception network and then a Fréchet distance is computed between their corresponding feature maps. We use the implementation of FID in CAGAN [2]. Specifically, we use 50K real images and 50K generated images to compute statistics, respectively. Perceptual Path Length (PPL) is proposed in StyleGAN [1] to measure the smoothness of latent space. We adopt the PPL implementation in CAGAN [2] for a fair comparison. PSNR and LPIPS are used by CAGAN to evaluate the image projection ability. A given real image is first mapped back to the latent space through optimizer such as L-BFGS. The projected image is obtained by feeding this resulting latent code to the generator. Then, the PSNR and LPIPS distance are computed between the projected image and the original image again. A smaller value indicates that the generator can model the distribution in real world better. We compute these two metrics using our own implementation.

2 Distillation without GAN Loss

In Section. 3.3 of the main paper, we highlight that the mapping network decides whether a student can learn similar output to that of the teacher. To further examine this hypothesis, we train the student in a fully supervised manner. Specifically, we remove the GAN loss and treat the z and $G_t(z)$ as input/label pairs to train the student network. The result is shown in Fig. 1. It shows that the

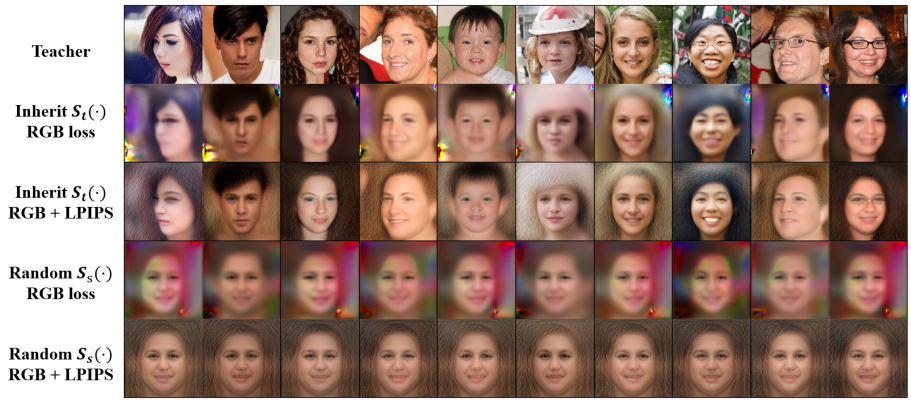


Fig. 1: Distillation without GAN loss.

Table 1: Ablation study about relation mimicking. Single View brings marginal improvement. Random Offset even has negative effect. Our LD loss consistently improves the performances of both RGB and RGB+LPIPS.

Mimicking Loss	$\mathcal{L}_{LD}$	FID
RGB	N/A	9.41
RGB + Random Offset	KL	9.80
RGB + Single View	KL	9.47
RGB + LD	L2	9.16
RGB + LD	KL	9.05
RGB + LPIPS	N/A	8.61
RGB + LPIPS + LD	L2	8.64
RGB + LPIPS + LD	KL	**8.26**

student cannot learn any meaningful content in the distillation process without a suitable mapping network. It yields the same face-like output for all the input noise.

3 Latent-Direction-Based Distillation Loss

The proposed latent-direction-based loss is essentially a relation loss. We are interested in whether the benefit brought by $\mathcal{L}_{LD}$ comes from relation mimicking or from the latent-direction-based augmentation. Specifically, we consider three variants: 1) Single View, namely the similarity is computed inside the normal samples rather than between normal samples and augmented samples, 2) Random Offset, namely we move w along a random direction to get f_i' instead of along the latent direction, 3) Our latent-direction-based method (abbreviated as LD).

4 Image Editing

We demonstrate the superiority of our method on image editing, including style mixing and interpolation. Given two real face images I_A, I_B, we first project them back to the latent space and get w_A, w_B. Both w_A and w_B are of shape $L \times D$, where L is the number of convolution layers and D is the dimension of latent code. For style mixing, we replace the $i-$th vector in w_A with that from w_B. We set $i \in [1, 3]$, $i \in [5, 8]$ and $i \in [10, 13]$ for coarse, middle and fine style mixing, respectively. For interpolation, we linearly combine the latent code with β controls the weight: $w = \beta \cdot w_A + (1 - \beta) \cdot w_B$, and then feed w into generator to get the interpolation results. We edit the images on resolution 256×256.

5 StyleGAN2 Linear Separability

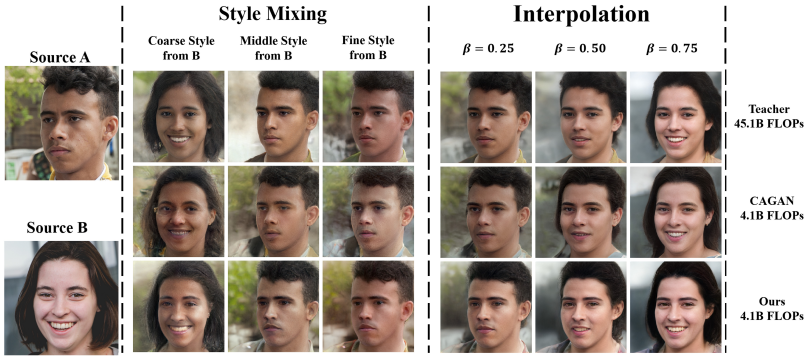
A well-trained StyleGAN2 model is linear separable in the latent space. An example is shown in Fig. 3. The results are shown in Fig. 2. For style mixing, CAGAN always has artifacts in face shape (coarse style) and skin color (middle shape). In contrast, the synthesized results of our method are more realistic and correspond better with two source images. In the coarse style case, our result corresponds well on face shape and facial components with source B. In the fine style case, our result corresponds well on lighting and skin color with source B. For interpolation, we also observe a smoother change than CAGAN, showing that our method learns a better structure in the latent space.

6 Image Projection

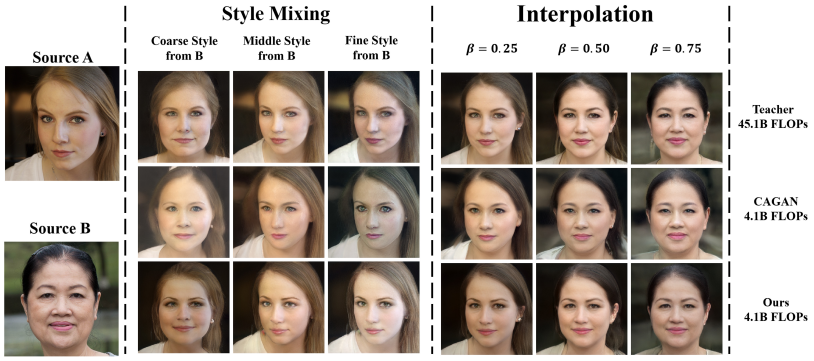
We show image projection results of our method in Fig. 4. All the real images come from Helen Set55 [2] and are not seen in the training stage. Our model reconstructs them with high quality.

7 Generation Results

We show more generation results of FFHQ and LSUN church datasets in Fig. 5 and Fig. 6, respectively.

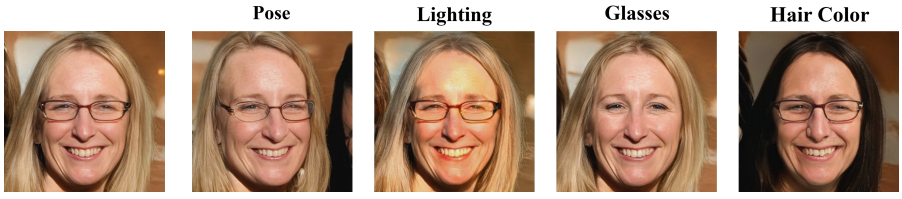


(a) In coarse style mixing, CAGAN generates glasses, which does not appear in both source images. CAGAN also produces blurry images in middle style case. In contrast, our style mixing results are more realistic and more similar to teacher.



(b) CAGAN generates lighting artifacts in coarse case and skin color artifacts in fine case, while our results are more realistic. In interpolation of CAGAN, the earrings disappear in $\beta = 0.25$ but appear again in $\beta = 0.50$. In contrast, our results are much smoother.

Fig. 2: Image editing results.



(a) Original (b) Single factor change

Fig. 3: StyleGAN2 shows good factorization in the w space. It is possible to control a single semantic factor such as pose, lighting condition, glasses and hair color by moving the style vector w of a certain layer along a specific direction.

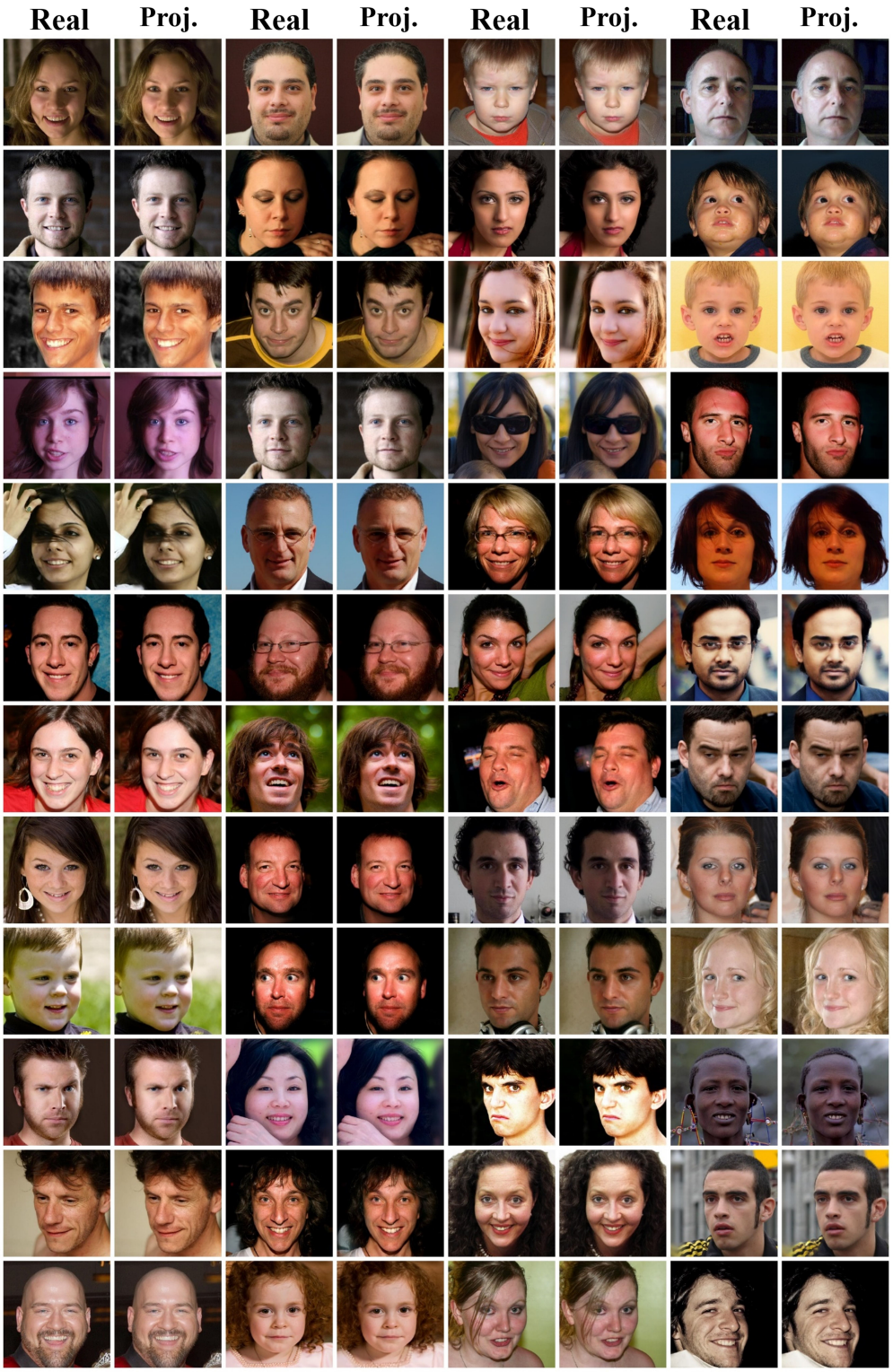


Fig. 4: Image projection results. In each pair, the left image is from real world (not from training set) and the right image is the projected result by our model. Our method can model the real face distribution well.

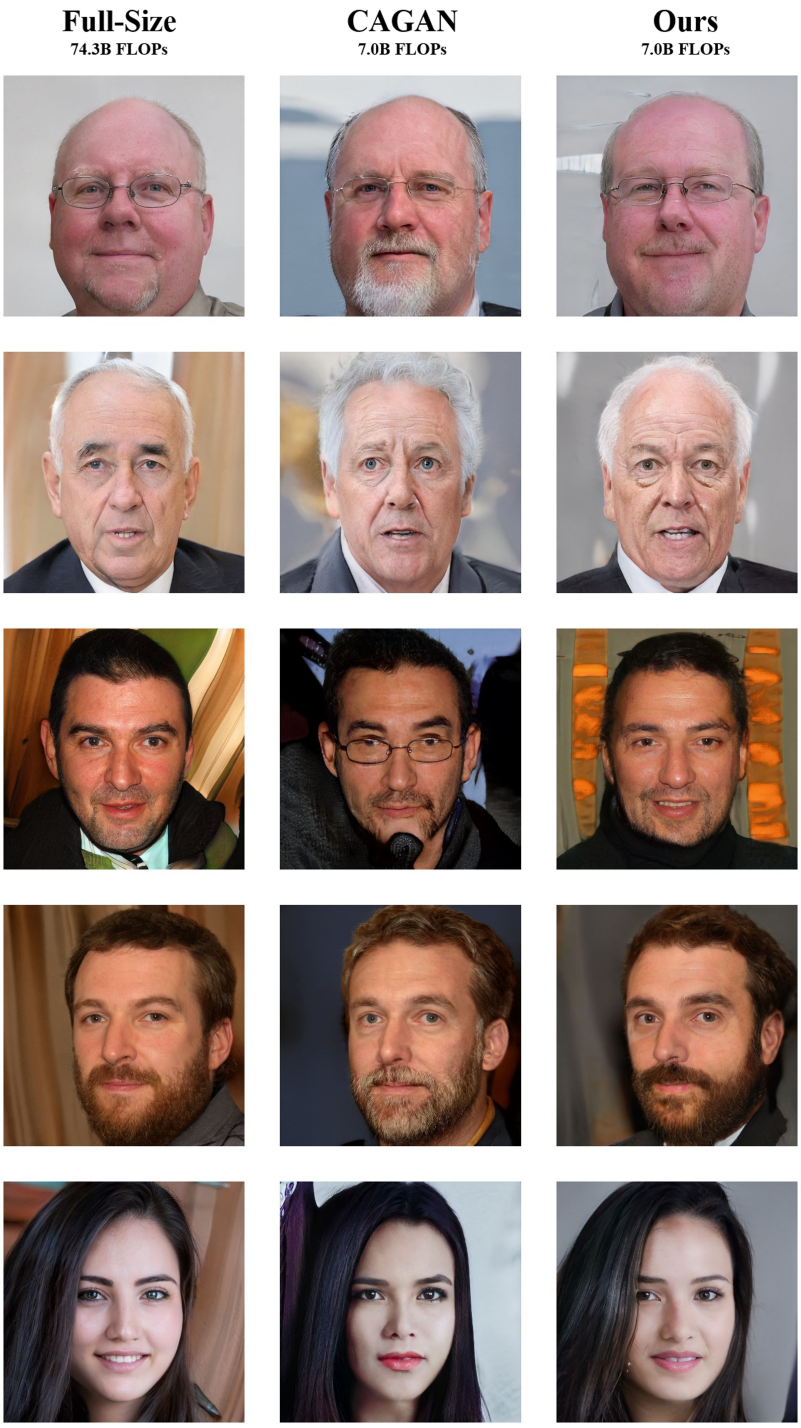


Fig. 5: Generation results on resolution 1024×1024. The synthesized images of Our method are of better quality than CAGAN. In several semantic factors such as beard, haircut and glasses, our results are more similar to the full-size model even though we do not inherit convolution weights.

Fig. 6: Generation results on LSUN church on resolution 256×256.

References

1. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
2. Liu, Y., Shu, Z., Li, Y., Lin, Z., Perazzi, F., Kung, S.Y.: Content-aware gan compression. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)