LaDiffGAN: Training GANs with Diffusion Supervision in Latent Spaces

Xuhui Liu[1]*, Bohan Zeng[1]*, Sicheng Gao [1], Shanglin Li[1], Yutang Feng[1],
Hong Li[1], Boyu Liu[1], Jianzhuang Liu[2],Baochang Zhang[1,3,4†]
[1]Beihang University [2]Shenzhen Institute of Advanced Technology, Shenzhen, China
[3]Zhongguancun Laboratory, Beijing, China [4]Nanchang Institute of Technology, Nanchang, China

In this supplementary material, we first describe the pretraining process of the latent diffusion models in Section A. We also provide the details of the other loss functions (mentioned in Section 3.4 in the main paper) in Section B. Then, we present more visual results on various datasets to further verify the effectiveness of LaDiffGAN in Section C. Finally, we state the ethical impact in Section E.

A. Pretraining of the Latent Diffusion Models

Diffusion Models. Following [1], the inference process p_θ of diffusion models, which denoises a normally distributed variable x_T to a target image x_0, can be formulated as:

$$\begin{aligned} p_\theta(\mathbf{x}_{0:T}) =& p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \\ p(\mathbf{x}_T) =& \mathcal{N}\left(\mathbf{x}_T \mid 0, I\right), \\ p_\theta(\mathbf{x}_{t-1}|x_t) =& \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)), \end{aligned} \tag{S1}$$

where $\mathbf{x}_1, ..., \mathbf{x}_T$ are latent features with added noise, $p_\theta(\mathbf{x}_{0:T})$ represents the joint distribution which performs the image generation process and is conducted as a reverse Markovian process with learnable Gaussian transitions $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$.

Besides, the forward process (Eq. 2 in the main paper), the inference (sampling) process (Eq. S1), and the optimization target (Eq. 5 in the main paper) are already mentioned.

Pretraining of the AutoEncoders. Let x_d and $x_{recon,d}$ be the real images and the reconstructed images of the target domain, and x_s and $x_{recon,s}$ be the real images and the reconstructed images of the source domain, respectively. We adopt the L1 loss and the VGG perceptual loss [3] to train our autoencoder $\mathcal{E}_d$ on the target domains. The L1 loss is introduced to alleviate the gap between the reconstructed images and the input images:

$$L_1 = |x_d - x_{recon,d}|. \tag{S2}$$

The VGG perceptual loss is adopted to align the high-level semantics of $x_{recon,d}$ with that of x_d. To be specific, we first utilize the pretrained VGG-19 [7] model f_{VGG} to extract L hierarchical features of $x_{recon,d}$ and x_d at 4 different scales, and then we use the L1 loss to penalize the differences between these features, which is formulated as:

$$L_{perceptual} = \sum_{j=1}^{4}\sum_{l=1}^{L} L_1(f_{VGG,l}(x_{d,j}), f_{VGG,l}(x_{recon,d,j})). \tag{S3}$$

Similarly, the autoencoder $\mathcal{E}_s$ of the source domain is optimized with x_s and $x_{recon,s}$ in the same manner.

B. Other Loss Functions

Adversarial loss. The adversarial loss $L_{GAN} = L_{GAN}^{s\to d} + L_{GAN}^{d\to s}$ is adopted to match the distribution of translated images to the target/source domain distribution:

$$\begin{aligned} L_{GAN}^{s\to d} &= \mathbb{E}_{x\sim X_d}[(\mathcal{D}_d(x))^2] + \mathbb{E}_{x\sim X_s}[(1-\mathcal{D}_d(\mathcal{G}_{s\to d}(x)))^2], \\ L_{GAN}^{d\to s} &= \mathbb{E}_{x\sim X_s}[(\mathcal{D}_s(x))^2] + \mathbb{E}_{x\sim X_d}[(1-\mathcal{D}_s(\mathcal{G}_{d\to s}(x)))^2]. \end{aligned} \tag{S4}$$

Cycle loss. In order to alleviate the mode collapse problem, the cycle consistency loss $L_{\text{cycle}} = L_{\text{cycle}}^{s\to d} + L_{\text{cycle}}^{d\to s}$ is applied to supervise the generators $\mathcal{G}_{s\to d}$ and $\mathcal{G}_{d\to s}$. We align $x \in X_s$ ($x \in X_d$) with the sequential translated results $x_{s\to d\to s}$ ($x_{d\to s\to d}$), so that the synthesis of the GAN model can be effectively supervised. Although the cycle loss may be too restrictive, it is the simplest and most effective way to ensure the quality of the generated results.

$$\begin{aligned} L_{\text{cycle}}^{s\to d} &= \mathbb{E}_{x\sim X_s}[|\ x - \mathcal{G}_{d\to s}(\mathcal{G}_{s\to d}(x))|_1], \\ L_{\text{cycle}}^{d\to s} &= \mathbb{E}_{x\sim X_d}[|\ x - \mathcal{G}_{s\to d}(\mathcal{G}_{d\to s}(x))|_1]. \end{aligned} \tag{S5}$$

Identity loss. We employ the identity loss $L_{identity} = L_{identity}^{s\to d} + L_{identity}^{d\to s}$ to ensure that the color distribution of the input and translated images are similar. The images

*These authors contributed equally.
†Corresponding Author: bczhang@buaa.edu.cn.

$x \in X_d$ ($x \in X_s$) should not be changed after being translated by the generator $\mathcal{G}_{d\to s}$ ($\mathcal{G}_{s\to d}$).

$$\begin{aligned} L_{identity}^{s\to d} &= \mathbb{E}_{x\sim X_d}[\mid x - \mathcal{G}_{s\to d}(x)\mid_1], \\ L_{identity}^{d\to s} &= \mathbb{E}_{x\sim X_s}[\mid x - \mathcal{G}_{d\to s}(x)\mid_1]. \end{aligned} \tag{S6}$$

CAM loss. We follow U-GAT-IT [4] and introduce the CAM loss $L_{CAM} = L_{CAM}^{s\to d} + L_{CAM}^{\mathcal{D}_d} + L_{CAM}^{d\to s} + L_{CAM}^{\mathcal{D}_s}$ to let $\mathcal{G}_{s\to d}/\mathcal{G}_{d\to s}$ and $\mathcal{D}_d/\mathcal{D}_s$ know what makes the most difference between two domains or where they need to improve:

$$\begin{aligned} L_{CAM}^{s\to d} &= -\left(\mathbb{E}_{x\sim X_s}[\log(\eta_s(x))] + \mathbb{E}_{x\sim X_d}[\log(1-\eta_s(x))]\right), \\ L_{CAM}^{\mathcal{D}_d} &= \mathbb{E}_{x\sim X_d}[(\eta_{\mathcal{D}_d}(x))^2] + \mathbb{E}_{x\sim X_s}[(1-\eta_{\mathcal{D}_d}(\mathcal{G}_{s\to d}(x))^2], \\ L_{CAM}^{d\to s} &= -\left(\mathbb{E}_{x\sim X_d}[\log(\eta_d(x))] + \mathbb{E}_{x\sim X_s}[\log(1-\eta_d(x))]\right), \\ L_{CAM}^{\mathcal{D}_s} &= \mathbb{E}_{x\sim X_s}[(\eta_{\mathcal{D}_s}(x))^2] + \mathbb{E}_{x\sim X_d}[(1-\eta_{\mathcal{D}_s}(\mathcal{G}_{d\to s}(x))^2], \end{aligned} \tag{S7}$$

where η_s, $\eta_{\mathcal{D}_d}$,η_d, and $\eta_{\mathcal{D}_s}$ are auxiliary classifiers (inspired by CAM [9]).

C. Visual Results

C.1. Visualization of Ablation Study

In this section, we compare our LaDiffGAN with UGATIT [4] and UGATIT trained by diffusion-GAN [8]. Besides, we provide the results of MUNIT [2], UNIT [6], and DRIT [5]. As shown in Fig. A, although Diffusion-GAN alleviates the unnatural details caused by UGATIT, our LaDiffGAN can generate more visually pleasing results with more consistent styles with the target domains.

C.2. More Visual Results

In this section, we provide more visual results of LaDiffGAN on the cat2dog, horse2zebra, and vangogh2photo datasets. The results are shown in Fig. B. We can observe that the compared methods encounter varying degrees of distortions, while our LaDiffGAN exhibits superior visual quality on all translation tasks. For instance, in the second row of Fig. B, the generated cat image of U-GAT-IT has an unnatural look with one eye larger than the other, whereas the generated result of LaDiffGAN is more realistic.

Moreover, we show more visualization results of LadiffGAN on the selfie2anime, cat2dog, and vangogh2photo datasets in Fig. C, Fig. D, and Fig. E. These results again validate the remarkable ability of LaDiffGAN in unsupervised image-to-image translation.

D. More Quantitative Results

We report the comparison results in terms of FID in Table 1. It shows that LaDiffGAN achieves better performance on the selfie2anime and anime2selfie tasks.

Table 1. Quantitative Comparisons in FID and KID.

Model	selfie2anime		anime2selfie	
	FID	KID	FID	KID
U-GAT-IT	94.5116	2.62 ± 0.30	111.573	1.57 ± 0.26
LaDiffGAN	**91.6198**	**1.97 ± 0.23**	**111.457**	**1.52 ± 0.31**
UNIT	116.813	5.14 ± 0.41	137.740	**4.04 ± 0.36**
UNIT+LaDiffGAN	**109.164**	**4.33 ± 0.44**	**137.739**	4.42 ± 0.39

E. Ethic Impact

This work can be used for the human face generation task which is common in mobile phone photography. It does not have a direct negative social impact. Because of personal security, we should prevent it from being abused for malicious purposes.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 1

[2] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 2

[3] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 1

[4] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwang Hee Lee. U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In *ICLR*, 2019. 2

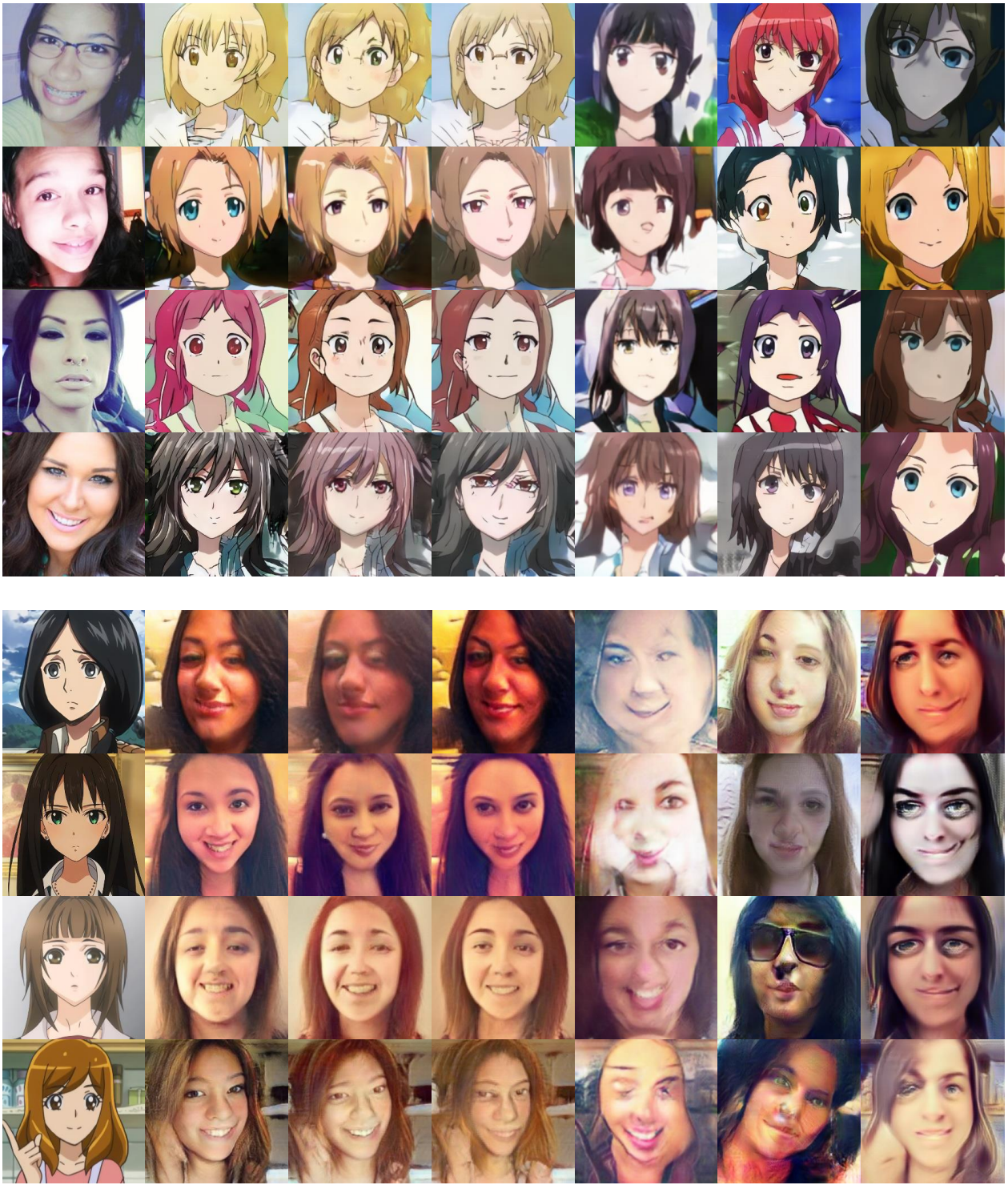
[5] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018. 2

[6] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NeurIPS*, 2017. 2

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[8] Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. *arXiv:2206.02262*, 2022. 2

[9] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 2



(a) Input **(b) LadiffGAN** **(c) DiffGAN** **(d) U-GAT-IT** **(e) UNIT** **(f) MUNIT** **(g) DRIT**

Figure A. Additional qualitative results on the selfie2anime dataset. The top 4 rows show some results of selfie2anime translation, while the bottom 4 rows present some results of anime2selfie translation.

Figure B. Additional qualitative results on the horse2zebra, cat2dog and vangogh2photo datasets.

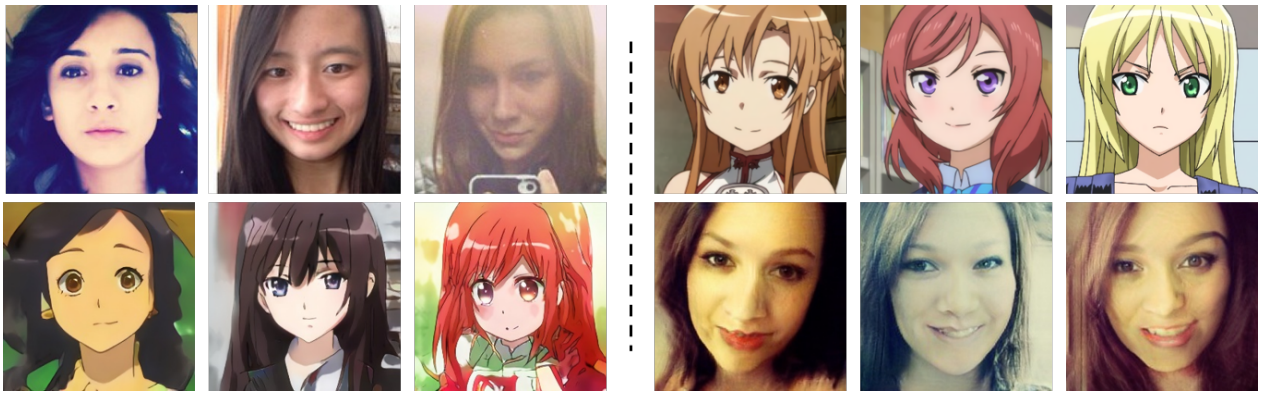


Figure C. Additional visualization of LaDiffGAN on the selfie2anime dataset. The first row is input, while the second row shows the generated results.

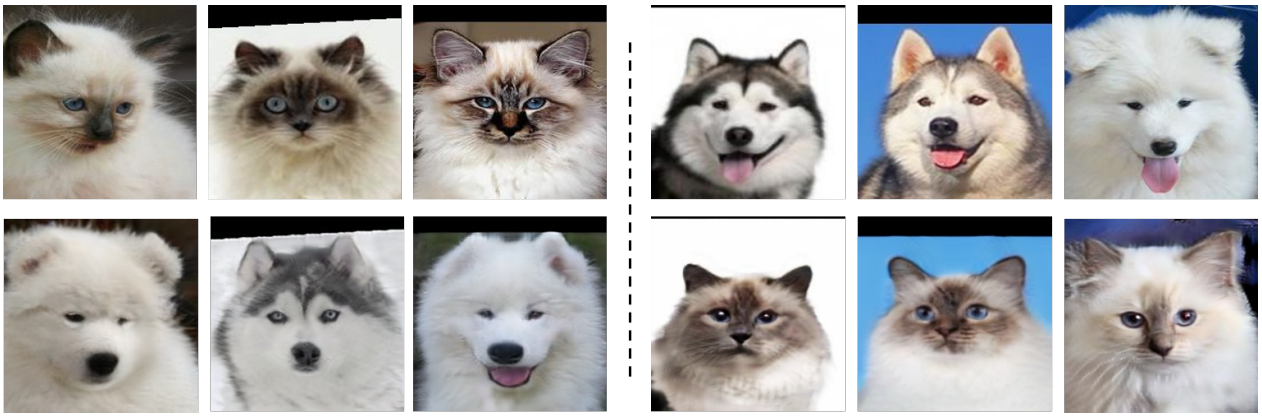


Figure D. Additional visualization of LaDiffGAN on the cat2dog dataset. The first row is input, while the second row shows the generated results.

Figure E. Additional visualization of LaDiffGAN on the vangogh2photo dataset. The first row is input, while the second row shows the generated results.