Phasic Content Fusing Diffusion Model with Directional Distribution Consistency for Few-Shot Model Adaption (Supplementary Material)

Teng Hu[1,*], Jiangning Zhang[2,*], Liang Liu[2], Ran Yi[1,†], Siqi Kou[1]
Haokun Zhu[1], Xu Chen[2], Yabiao Wang[2,3], Chengjie Wang[1,2], Lizhuang Ma[1]
[1]Shanghai Jiao Tong University [2]Youtu Lab, Tencent [3]Zhejiang University
{hu-teng, ranyi, happy-karry, zhuhaokun, ma-lz}@sjtu.edu.cn;
{vtzhang, leoneliu, cxxuchen, caseywang, jasoncjwang}@tencent.com;

A. Overview

This supplementary material consists of:

- The implementation details on the training and testing process (Sec. B);
- Comparison with SDEdit. (Sec. C)
- Ablation study on the phasic factor T_s in $m(t)$, the parameters in ICSG and DDC loss (Sec. D);
- The theoretical analysis and more experiments on our iterative cross-domain structure guidance strategy (ICSG). (Sec. F)
- The theoretical analysis of our directional distribution consistency loss. (Sec. G)

B. Implementation Details

B.1. Training Details

In our diffusion model [6], we set the maximum step to be 1000. We set the phasic factor T_s in $m(t) = \frac{1}{1+e^{-(t-T_s)}}$ to 300 and the parameter α in $w(t) = 1 - (\frac{t}{T})^{\alpha}$ to 3. We start training from a pre-trained diffusion model with cosine noise schedule [10] on the source dataset, and fine-tune it with our phasic content fusing strategy and corresponding loss functions. Using the pre-trained Unet network, we first train our phasic content fusion model with only the diffusion loss $\mathcal{L}_{dif}$ on the source dataset, with a batch size of 8 and a learning rate of $1e^{-4}$ for 1000 iterations, to avoid interference from random weights in the early training stage.

After training the phasic content fusion module, we train the entire model with the final loss function $\mathcal{L}$ (Equation (7) in the main paper), with a batch size of 8 and a learning rate of $1e^{-4}$. We set the hyperparameters λ_{DDC} and λ_{style} to 1.

*Equal contributions.
†Corresponding author.

Method	FFHQ → Sketches				FFHQ → Sketches			
	FID	IS ↑	IC-L ↑	SCS ↑	FID	IS ↑	IC-L ↑	SCS ↑
SDEdit (400)	82.14	1.95	0.43	0.47	154.99	1.85	0.45	0.50
SDEdit (500)	77.33	1.90	0.40	0.40	144.42	1.91	0.43	0.47
SDEdit (600)	70.96	1.88	0.38	0.33	137.79	1.93	0.37	0.44
PCF Only	57.62	2.11	0.52	0.51	137.79	2.70	0.60	0.69
Full Model	47.42	**2.36**	**0.56**	**0.62**	**119.65**	**3.41**	**0.63**	**0.84**

Table 1. **Comparison results between our model and SDEdit** with different nosing steps (400, 500 and 600) on FID, IS, IC-LPIPS and SCS metrics.

B.2. Testing Details

After training, we test our model with our iterative cross-domain structure guidance (ICSG) strategy. For the style enhancement factor K in ICSG, we set $K = 2$ for FFHQ [8] → Sketch [13], and $K = 1$ otherwise. Furthermore, for an input image x, we add 800-step noise into it as the starting point x_M, and employ ICSG in the denoising step until the stop step t_{stop} ($t_{stop} = 500$ for FFHQ [8] and $t_{stop} = 200$ for LSUN Church [14]). Note that a wide range of the stop step t_steop and style enhancement factor K have a good performance in few-shot domain adaption as illustrated in Sec. D.2. We only choose a relatively better parameter setting in the testing stage.

C. Comparison with SDEdit

SDEdit [9] is a model that maintains the content information during domain adaption by adding a t-step noise into a source image and denoise it. In comparison, our model utilize phasic content fusion (PCF) module to keep the content information. Different from SDEdit which only keeps the content information in the noised image and has no further contents injected during the denoising stage, our PCF constantly fuses the images in the denosing process with the features from source image using a three-layer convolution network, thereby aiding our model in autonomously acquir-

Figure 1. **Ablation study on T_s, the phasic factor in $m(t)$ on FFHQ → Cartoon [12].** When $T_s = 800$, the model only learns the target-domain details, ignoring the global style. When $T_s = 0$, the model learns the style and content information in the whole process, which leads to failed style transfer at t-small, causing an unstable training which generates artifacts and rough details in the output images.

ing content information from the original images. To further validate the effectiveness of our PCF, we compare our PCF with SDEdit in content preservation and generation quality.

In addition to the Inception Score (IS), Structural Consistency Score (SCS), and Intra-cluster pairwise LPIPS distance (IC-LPIPS) metrics employed in the main paper, we also incorporate the Fréchet Inception Distance (FID) [5] to measure the similarity between the features of the source data and the generated data according to their mean values and covariance. A lower FID suggests generated data is similar to source data with high diversity and realism.

In order to facilitate a more effective comparison between PCF and SDEdit, we refrain from utilizing the ICSG module for contour preservation (PCF only) and substitute our PCF by SDEdit with different noising steps. We compute the FID, IS, IC-LPIPS and SCS scores in Tab. 1, where the noising step of SDEdit ranges from 400 to 600 (the recommended parameter in its paper). It can be seen that our PCF outperforms SDEdit in terms of generation quality and diversity.

D. Ablation Study

D.1. Ablation Study on The Phasic Factor T_s

The phasic factor T_s in $m(t)$ is an important parameter that influences the generated results. A large T_s leads to the failure in style transfer since there are only $M - T_s$ steps to transfer the style. Similarly, a small T_s leads to failure in capturing target-domain details, causing rough details in the generated images. Moreover, when t is too small, the failure in style transfer also leads to an unstable training process which generates artifacts in the output images. To validate this, we conducted an ablation study on the phasic factor T_s and show the results in Fig. 1.

It can be seen that when $T_s = 800$, the model only transfers the local details in the target domain, ignoring the

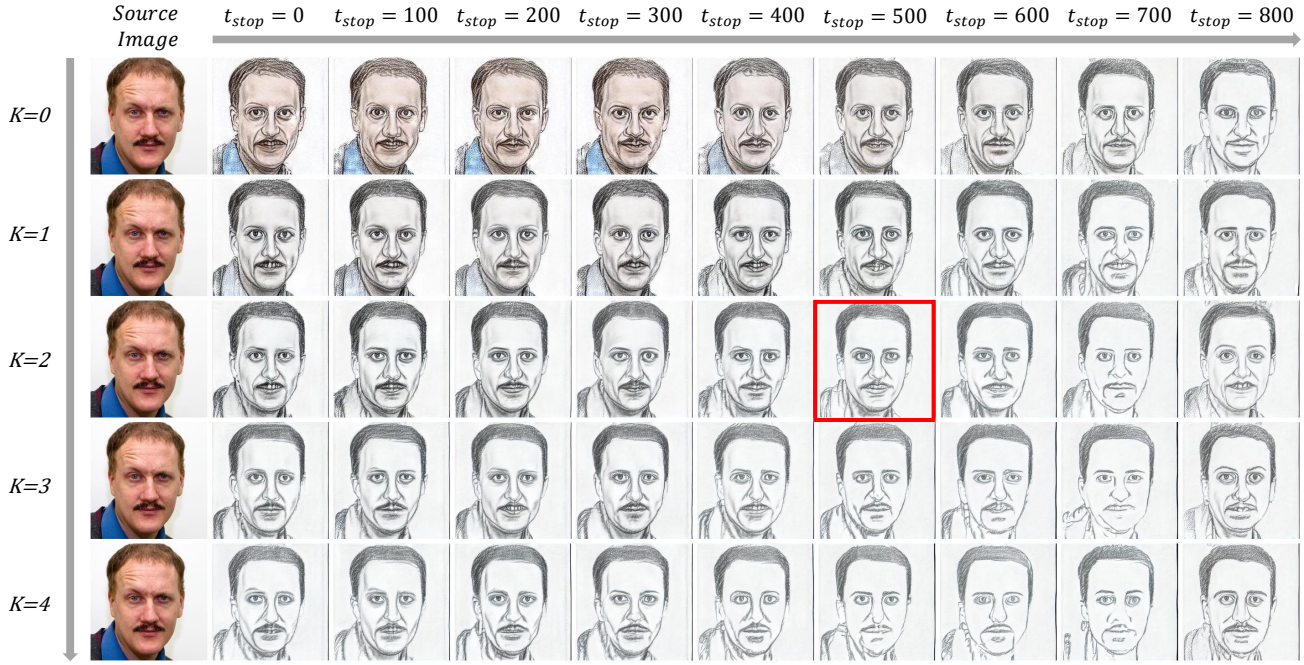

Figure 2. **Ablation Study on the stop step t_s and the repeating factor K in style enhancement module** with the filtering factor $N = 8$. As t_{stop} or K grows, the generated image shares less contents with the source image. When t and K is small, the model cannot eliminate the influence of the source image, i.e., generating wrong color in the output image.

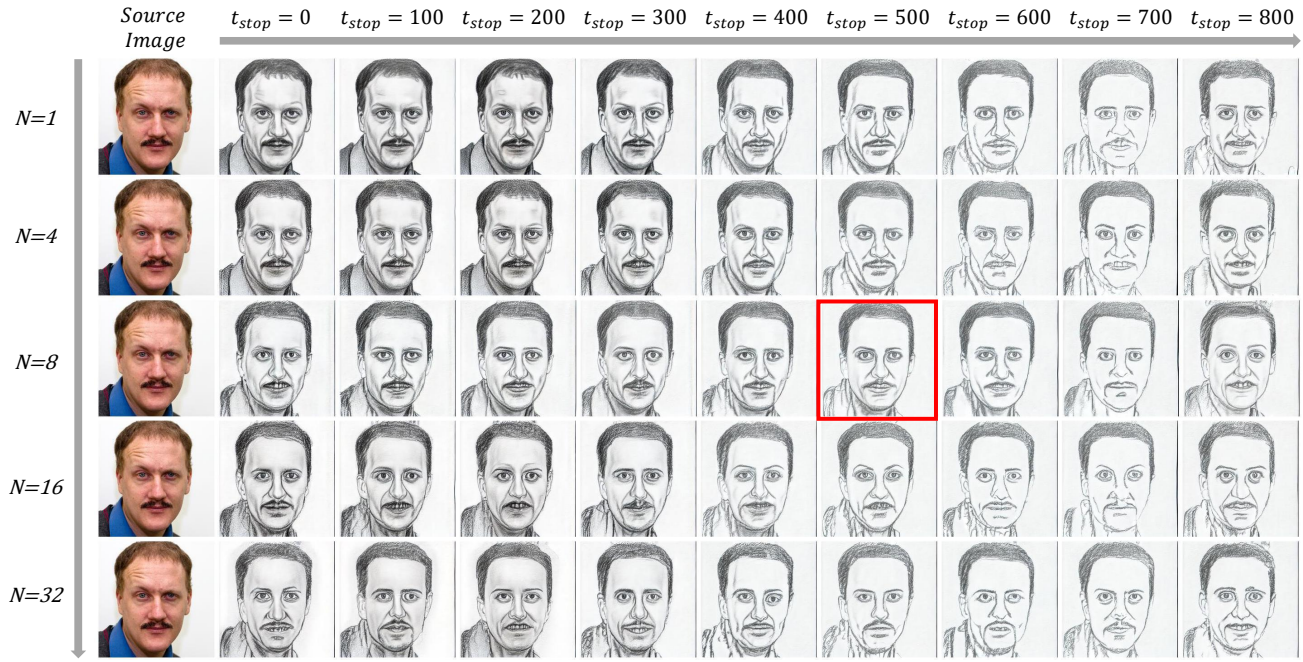

Figure 3. **Ablation Study on the stop step t_s and the filtering factor N** with the repeating factor $K = 2$. As t_{stop} or N grows, the generated image shares less contents with the source image.

global style. When $T_s = 0$, the model suffers from generating many artifacts, and the generated images all seem to be rough. Thus, we chose $T_s = 300$ as our default setting, which balances both style transfer and detail capturing.

Moreover, we also compute the FID scores between the cartoon dataset and our generated data with different T_s. The results are shown in Tab. 2. It can be seen that when T_s ranges from 200 to 400, the FID scores are similar, indicating that a wide range of T_s result in a good performance.

T_s	0	100	200	300	400	500	600	700	800
FID	136.51	137.00	125.46	**119.65**	123.75	145.62	157.11	161.40	155.93

Table 2. **Ablation study on** T_s, the phasic factor in $m(t)$ on FFHQ → Cartoon [12]. We evaluate the FID between the generated images and the cartoon dataset. It can be seen that when T_s ranges from 200 to 400, the FID scores are similar, indicating that a wide range of T_s result in a good performance.

Model	Ours	IDC loss	RSSA loss	NADA loss
Sketches	**47.42**	146.32	125.77	88.76
Cartoon	**119.65**	180.28	171.99	144.57

Table 3. **Quantitative comparison between our DDC loss and the losses in IDC, RSSA and StyleGAN-NADA.**

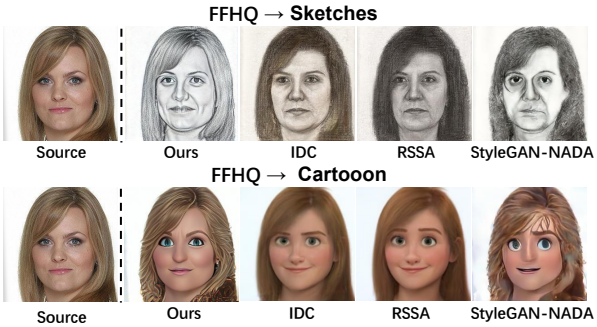

Figure 4. **Qualitative comparison results between our DDC loss and the losses in IDC, RSSA and StyleGAN-NADA.**

D.2. Ablation Study on ICSG

Our iterative cross-domain structure guidance strategy (ICSG) comprises three key parameters: the repeating factor K of the style enhancement module, the filtering factor N, and the stop step t_{stop}. To demonstrate a clear comparison among different parameter values, we conducted an experiment on the FFHQ → Cartoon task.

Firstly, we investigate the influence of the repeating factor K and stop step t_{stop} on the output of ICSG, as shown in Fig. 2 with $N = 8$. Next, we examine the impact of the filtering factor N and stop step t_{stop} on the output of ICSG, as illustrated in Fig. 3 with $K = 2$. (It should be noted that the default setting in our method is $K = 2$, $N = 8$, and $t_{stop} = 500$ here) We observed that as K, t_{stop}, or N increases, the model captures more style in the target domain but loses more content information and local structures. When both K and t_{stop} are small, the model cannot effectively eliminate the influence of the source image in terms of original color and texture. In summary, a bigger K and t_{stop} enhance the stylization effect and a smaller K, t_{stop} and N keep more content information. In general, a wide range of parameter values near our default setting ($K = 2$, $N = 8$, and $t_{stop} = 500$) yield favorable outcomes as illustrated in Fig. 2 and 3, indicating that our model is not too sensitive to the parameters.

E. More Ablation Study on DDC Loss

We compare our DDC loss with the losses in IDC [11], RSSA [16] and StyleGAN-NADA [3]. For a fair comparison, we exclusively substitute the DDC loss in our model with their losses, and keep the other conditions unchanged. The comparison results are shown in Tab. 3 and Fig. 4. It can be seen that our DDC loss outperforms the other distribution-consistency losses in diffusion-based few-shot domain adaption.

F. Details of Iterative Cross-domain Structure Guidance (ICSG)

F.1. Far More Than Few-shot Image Translation

In our main paper, we introduce a novel iterative cross-domain structure guidance strategy (ICSG) for image sampling, which helps to retain structural information. The proposed ICSG is not limited to few-shot image translation tasks but can be applied to any image-to-image translation task on any source and target domains. In this section, we aim to demonstrate the effectiveness of ICSG and show more experiments on image-to-image translation.

F.2. Derivation of ICSG

In this section, we provide theoretical proof derivations to explain why our method is effective. For the sake of convenience, we define the following notations:

Definition 1 *We define the denoising process* Θ_t *as:*

$$\Theta_t : \mathbf{R}^D \longrightarrow \mathbf{R}^D$$

$$x_t \mapsto x_{t-1}$$

$$\Theta_t(x_t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta) + \sigma_t z \sim p_\theta(x_{t-1}|x_t) \ ,$$

where D *is the dimension of the image, and* z *is a random variable from standard normal distribution.*

Definition 2 *We define the forward process* Φ_t *as:*

$$\Phi_t : \mathbf{R}^D \longrightarrow \mathbf{R}^D$$

$$x_0 \mapsto x_t$$

$$\Phi(x_0) = x_t(x_0, \epsilon) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \sim q(x_t|x_0) \ .$$

Definition 3 *We define the backward process* Ψ_t *as:*

$$\Psi_t : \mathbf{R}^D \longrightarrow \mathbf{R}^D$$

$$x_t \mapsto \hat{x_0}$$

$$\Psi_t(x_t) = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta) \sim p_\theta(\hat{x_0}|x_t).$$

Here, x_0 denotes the output image in the target domain, and y_0 denotes the source-domain image with the target structure $\phi_N(y_0)$. Our ICSG can be defined as follows:

$$ICSG(x_{t-1}|x_t, y_0) = \Theta_t(x_t) + \phi_N(SE(y_0)) - \phi_N(\Theta_t(x_t))$$
$$where\, SE(y_0) = \Theta_t \circ (\Phi_t \circ \Psi_t)^n \circ \Phi_t(y_0)\ ,$$

where $(\Phi_t \circ \Psi_t)^n$ is the style enhancement module. We have the following theorem:

Theorem 1 *With our ICSG, the generated image x_0 shares the same structure with the reference image y_0.*

$$\mathbf{E}_{x_0 \sim p_{t(data)}}(\phi_N(x_0)) = \mathbf{E}_{y_0 \sim p_{s(data)}}(\phi_N(y_0))\ , \quad (1)$$

where $p_{t(data)}$ is the target data distribution, and $p_{s(data)}$ is the source data distribution. This indicates that the structure of the output image x_0 is the same as that of the reference image y_0.

Proof 1. When t is small, $\Psi_{t-1}(x_{t-1})$ is very close to x_0 and $\phi_N(x)$ further blurs them. Thus, we can approximate $\phi_N(x_0)$ as $\phi_N(\Psi_{t-1}(x_{t-1}))$.

$$\begin{aligned}
&\mathbf{E}_{x_0 \sim p_{t(data)}}(\phi_N(x_0)) \\
&\approx \mathbf{E}(\phi_N(\Psi_{t-1}(x_{t-1}))) \\
&= \mathbf{E}(\phi_N(\Psi_{t-1}(\Theta_t(x_t) + \phi_N(SE(y_0)) - \phi_N(\Theta_t(x_t))))) \\
&= \mathbf{E}(\phi_N(\Psi_{t-1}(\Theta_t(x_t) - \phi_N(\Theta_t(x_t))))) \cdots PartI \\
&+ \mathbf{E}(\phi_N(\Psi_{t-1}(\phi_N(SE(y_0))))) \cdots PartII
\end{aligned}$$

Regarding $PartI$, we can utilize the linear properties of ϕ_N and Ψ_{t-1}, which yields:

$$\begin{aligned}
PartI =& \mathbf{E}(\phi_N(\Psi_{t-1} \circ \Theta_t(x_t) - \Psi_{t-1}(\phi_{N_s}(\Theta_t(x_t))))) \\
=& \phi_N(\mathbf{E}(\Psi_{t-1}(\frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta) + \sigma_t z)) \\
&- \mathbf{E}(\Psi_{t-1}(\phi_{N_s}(\Theta_t(x_t))))) \\
=& \phi_N(\mathbf{E}(\frac{1}{\sqrt{\alpha_t}}\Psi_{t-1}(x_t) - \frac{1}{\sqrt{\alpha_t}}\Psi_{t-1}(\phi_{N_s}(x_t)))) \\
=& \frac{1}{\sqrt{\alpha_t}}\phi_N(\mathbf{E}(\Psi_{t-1}(x_t) - \Psi_{t-1}(\phi_{N_s}(x_t)))) \\
\approx& 0\ ,
\end{aligned}$$

note that $\mathbf{E}(\epsilon_\theta) = 0$ and $\mathbf{E}(z) = 0$. Furthermore, the last approximation holds, given that α_t is close to 1 when t is small and our filtering factor N is not large (we have set N to be 8).

Regarding Part II, when t is small, $\Psi_{t-1}(y_{t-1})$ is in close proximity to y_0, and $\phi_N(x)$ further blurs them. Therefore, we have:

$$\Psi_{t-1}(\phi_N(SE(y_0)) = \hat{y_0}$$
$$PartII = \mathbf{E}(\phi_N(\hat{y_0})) \approx \mathbf{E}(\phi_N(y_0))\ .$$

Thus, we have demonstrated that:

$$\mathbf{E}_{x_0 \sim p_{t(data)}}(\phi_N(x_0)) = \mathbf{E}_{y_0 \sim p_{s(data)}}(\phi_N(y_0))\ . \quad (2)$$

F.3. Algorithm

The process of our ICSG can be summarized in Alg. 1.

Algorithm 1 ICSG for image-to-image translation

Input: Source image x and reference image y_0
Output: Generated image x_0

$x_M \sim q(x_M|x)$
for $t = M, ..., 1$ **do**
  $z \sim \mathcal{N}(0, I)$
  if $t \geq t_{stop}$ **then**
    $y_t \sim q(y_t|y_0)$
    for $i = 1, ..., K$ **do**
      $\hat{y_0} \sim p_\theta(\hat{y_0}|y_t)$
      $\hat{y_t} \sim q(\hat{y_t}|y_0)$ ▷ *Style Enhancement*
      $y_t \leftarrow \hat{y_t}$
    end for
    $y'_{t-1} \sim p_\theta(y_{t-1}|y_t)$
    $x'_{t-1} \sim p_\theta(x_{t-1}|x_t)$
    $x_{t-1} \leftarrow x'_{t-1} + \phi_N(y'_{t-1}) - \phi_N(x'_{t-1})$
  end if
  $x_{t-1} \sim p_\theta(x_{t-1}|x_t)$
end for
return x_0

F.4. Comparison on Image-to-image Translation

In this section, we present additional experimental results on the image-to-image translation task. Specifically, we compare our proposed method with two existing diffusion-based image-to-image translation methods, namely EGSDE [15] and ILVR [1], on cat-to-dog [2], male-to-female [7] and wild-to-dog [2] image-to-image translation task. To ensure a fair comparison, we use the pretrained diffusion model provided in the source code of EGSDE for all experiments and set the default parameters for both EGSDE and ILVR.

Fig. 5 shows the image translation results obtained by our proposed ICSG method. It can be seem that our model performs well in terms of both domain translation and structure preservation.

To provide a more comprehensive comparison, we also compare our method with state-of-the-art methods EGSDE [15] and ILVR [1]. We use the pretrained diffusion model as a baseline, which adds 700-step noise to x_0 and denoises it. For ILVR, we use filtering factors N of 4 and 32, which were effective in their paper. Moreover, since none of our method, ILVR, nor the fine-tuned model relies on additional classifiers, we conduct experiments on EGSDE with and without a classifier separately. The comparison results are shown in Fig. 6. The fine-tuned model loses much of the structural information after denoising. As for ILVR, when the filtering factor $N = 32$, the trans-

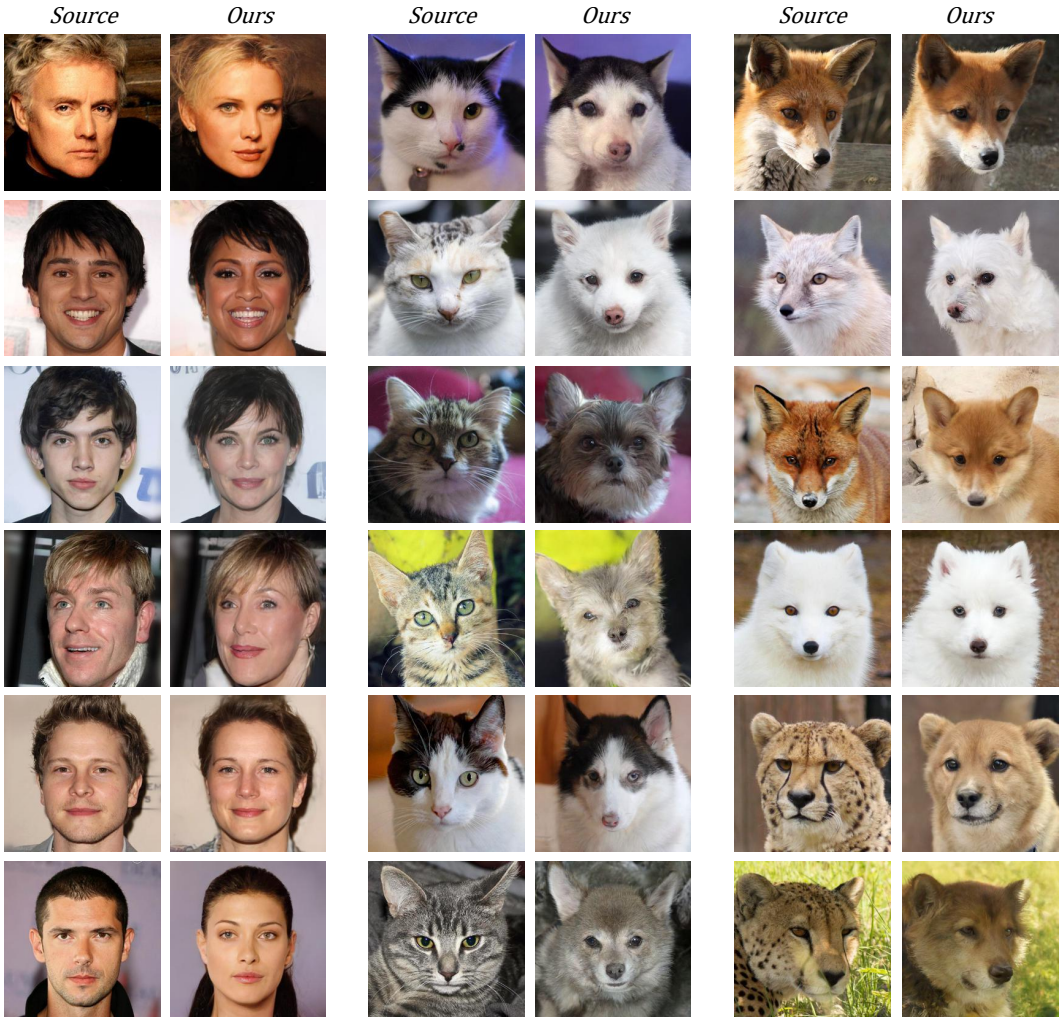

Figure 5. Our ICSG results on male-to-female, cat-to-dog and wild-to-dog image-to-image translation. The generated results achieve good performance in both structure preservation and domain translation.

lated images lose much structural information as well, and when $N = 4$, it tends to reconstruct the source images. There is no distinct difference between the two EGSDE results with and without a classifier. However, both of them lose some structural information in the generated images. In contrast, our model achieves good performance in both structure preservation and image translation.

G. Analysis on Directional Distribution Consistency Loss

Our directional distribution consistency loss explicitly constrains the centrality consistency between the generated distribution and the target distribution, while preserving the structural consistency of the generated distribution and the original distribution. In this section, we provide a theoretical analysis to demonstrate that the prior loss functions in the existing few-shot image generation tasks share similar goals with our approach, but they suffer from the distribution rotation problem, which can cause unstable training and low training efficiency.

IDC [11] proposes a cross-domain distance consistency loss that can maintain the structure of the generated distribution and prevent overfitting. Based on this, RSSA [16] further designs a cross-domain spatial structural consistency loss that can solve the drift problem of the generated samples in the target domain. However, both methods lack a deep analysis of the loss functions and suffer from distribution rotation during the training process. For the sake of convenience in our derivation, we use the loss function of IDC as an example for demonstration (since RSSA only addresses the issue of distribution drift, its proof follows a similar process).

To make our analysis clearer, we denote x and y as the

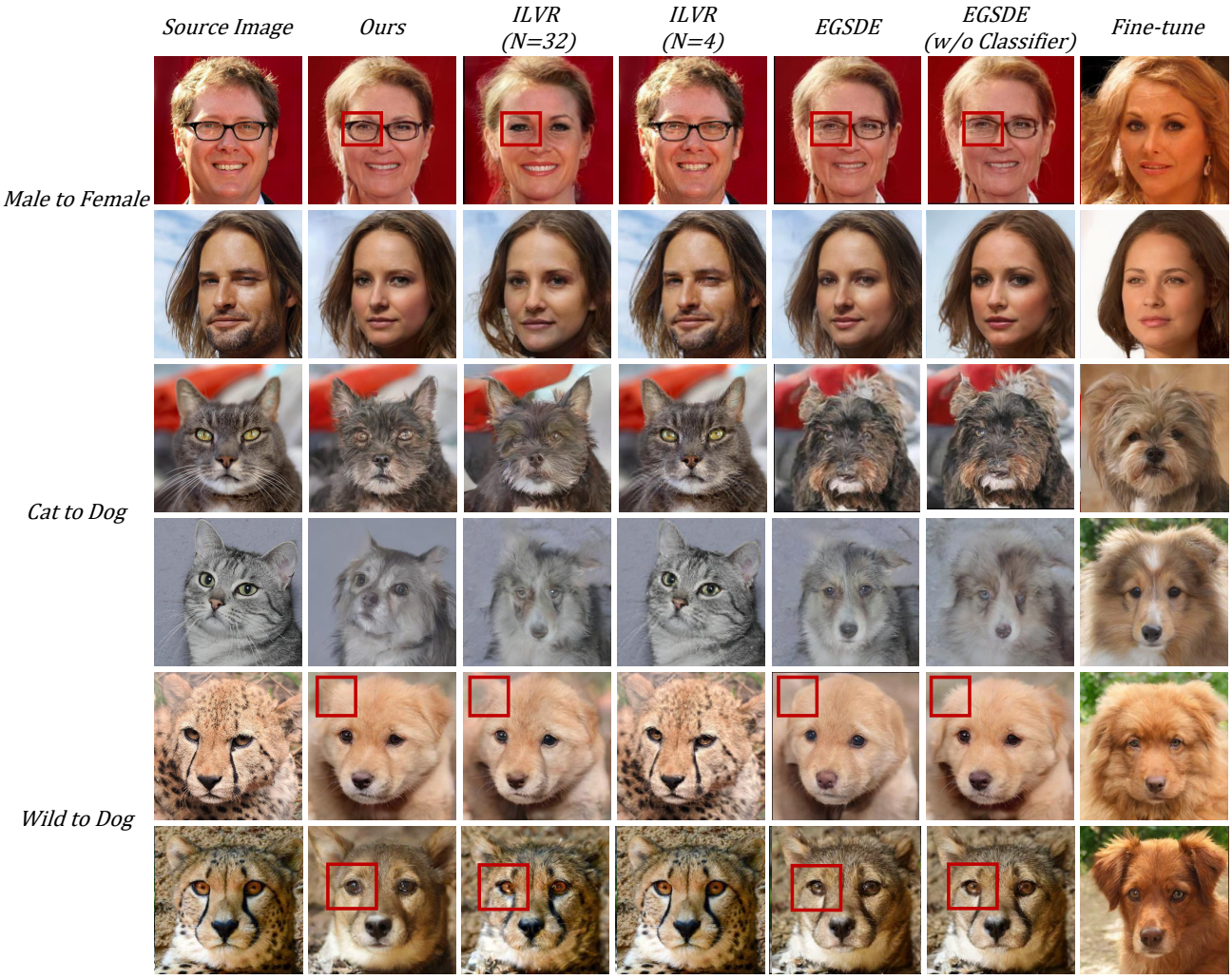

Figure 6. Comparison between our method and existing methods based on diffusion model in domain transfer. From lest to right, the images are: images from source domain, Our model, ILVR with filtering factor $N = 32$, ILVR with filtering factor $N = 4$, EGSDE, EGSDE without classifier and the fine-tuned model.

latent variables, $P_S(x)$, $P_T(x)$ and $P_G(x)$ as the source, target and generated distribution, $S(x)$, $T(x)$ and $G(x)$ as the corresponding images of the latent variable x in the source, target and generated distribution, and $C(\cdot)$ as the distribution center.

To minimize the few-shot loss in IDC, it satisfies:

$$cos(S(x_i), S(x_j)) = cos(G(x_i), G_(x_j)) \quad \forall x_i, x_j \in P_S(x) \,. \tag{3}$$

Thus, the structure of the generated domain is fixed. The generated distribution $P_G(x)$ can only rotate or move along the axis that crosses the origin and distribution center $C(G)$ (note that when moving the distribution along the axis, the scale of the distribution also varies.). So, the major concern is to determine where the center of the generated distribution $C(G)$ is located. Based on our analysis, we present the following theorem:

Theorem 2 *The center of the generated distribution $C(G)$ coincides with that of the target distribution $C(T)$ with the adversarial loss and few-shot loss:*

$$\int_x G(x)P_G(x)dx = \int_x T(x)P_T(x)dx \,. \tag{4}$$

Proof 2 The adversarial loss in GANs[4] is to find a generator G that satisfies $P_G(x) = P_T(x)$, which can be transferred into: $G(x) = T(x)$, $\forall x \sim P_T(x)$. In high-dimensional space, we can employ cosine distance to measure the similarity. Then, we rewrite the goal of the adversarial loss as:

$$G = \underset{G}{argmin}\, \mathbf{E}_{x\sim P_T(x)} |cos(G(x), T(x)) - 1| \,.$$

According to Eq.(3), we can also rewrite the goal of the few-shot loss functions as:

$$G = \underset{G}{argmin}\, \frac{1}{2}\mathbf{E}_{x,y\sim P_S(x)} |cos(S(x), S(y)) - cos(G(x), G(y))| \,.$$

Combining both the adversarial and few-shot loss together, we have the final optimization goal:

$$G = \underset{G}{argmin}\, \mathbf{E}_{x\sim P_T(x)}|cos(G(x), T(x)) - 1| + \frac{1}{2}\mathbf{E}_{x,y\sim P_S(x)}|cos(S(x), S(y)) - cos(G(x), G(y))| \,. \tag{5}$$

In high-dimensional space, any tow points has almost the same Euclidean distance, indicating the the modulus of the each vector are extremely close, denote the modulus as $\sqrt{\lambda}$. So, we transform Eq.(5) into:

$$G = \underset{G}{argmin} \int_x |G(x)T(x) - \lambda| P_T(x)dx + \frac{1}{2}\int_x\int_y |S(x)S(y) - G(x)G(y)| P_G(x)P_G(y)dxdy \,. \tag{6}$$

Taking the gradient on G in Eq.(6) and with the symmetry property of x and y, the optimal G^* satisfies:

$$\begin{aligned} &\int_x T(x)P_T(x)dx - \int_x\int_y G(x)P_G(x)P_G(y)dxdy = 0 \\ \Longleftrightarrow\ &\int_x T(x)P_T(x)dx = \int_x G(x)P_G(x)dx \int_y P_G(y)dy \\ \Longleftrightarrow\ &\int_x G^*(x)P_{G^*}(x)dx = \int_x T(x)P_T(x)dx \,. \end{aligned} \tag{7}$$

The optimal generation distribution aligns with the center of the target distribution (as shown in Eq.(7)). Once the distribution center is fixed, the generated distribution cannot shift along the axis that passes through the origin and the center. Therefore, the scale of the generated distribution matches that of the source distribution. Unfortunately, this does not solve the issue of distribution rotation, which can result in an unstable and ineffective training process.

In contrast, our directional distribution consistency loss maintains the distribution center and structure explicitly and any rotation or shift of the generated distribution result in an increase of our loss function.

References

[1] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021. 5

[2] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020. 5

[3] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022. 4

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 7

[5] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 2

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1

[7] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 5

[8] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1

[9] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 1

[10] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 1

[11] Utkarsh Ojha, Yijun Li, Jingwan Lu, Alexei A Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10743–10752, 2021. 4, 6

[12] Justin NM Pinkney and Doron Adler. Resolution dependent gan interpolation for controllable image synthesis between domains. *arXiv preprint arXiv:2010.05334*, 2020. 2, 4

[13] Xiaogang Wang and Xiaoou Tang. Face photo-sketch synthesis and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(11):1955–1967, 2008. 1

[14] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 1

[15] Min Zhao, Fan Bao, Chongxuan Li, and Jun Zhu. Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations. *arXiv preprint arXiv:2207.06635*, 2022. 5

[16] Jingyuan Zhu, Huimin Ma, Jiansheng Chen, and Jian Yuan. Few-shot image generation with diffusion models. *arXiv preprint arXiv:2211.03264*, 2022. 4, 6