DiffusionCLIP: Text-Guided Diffusion Models for Robust Image Manipulation

Jun Hyung Lee

Korea University Department of Computer Science and Engineering

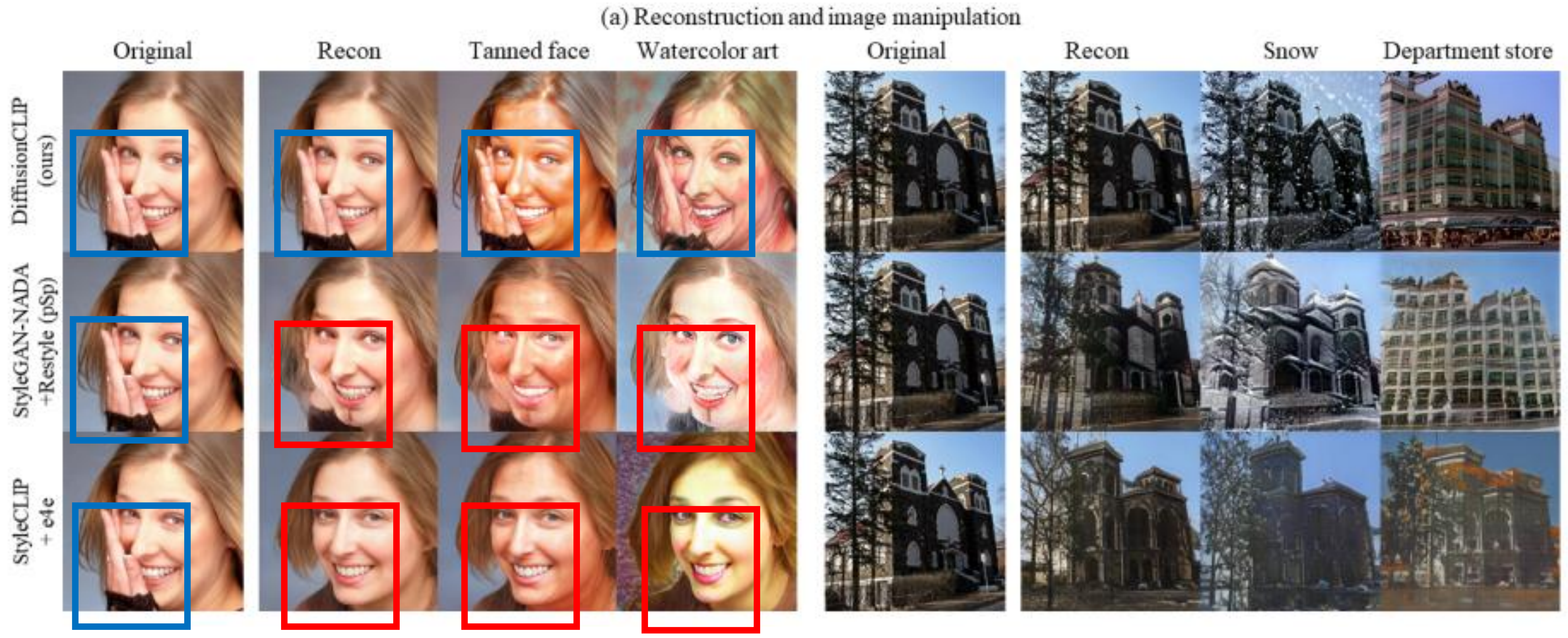
IE LAB Interlligence Engineering

References

- **Learning Transferable Visual Models From Natural Language Supervision,** Alec Radford (2021)
- **StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery,** Or Patashnik (2021)
- **StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators,** Rinon Gal (2021)
- **Diffusion Models Beat GANs on Image Synthesis,** Prafulla Dhariwal (2021)
- **Denoising Diffusion Implicit Models**, Jiaming Song (2021)
- **Denoising Diffusion Probabilistic Models**, Jonathan Ho (2020)
- **Denoising Diffusion-based Generative Modeling: Foundations and Applications,** Karsten Kreis, Ruiqi Gao, Arash Vahdat (CVPR Diffusion Tutorial 2022)
- **https://colab.research.google.com/drive/1E8QHZ3BbkF6hzk0rRKzhfkySmYf_BZaE?usp=sharing#scrollTo=1pIycQuLBp6g** (Colab Tutorial)
- **https://www.youtube.com/watch?v=ZKAXgg_OAE0&ab_channel=**고려대학교산업경영공학부DSBA연구실
- **https://www.youtube.com/watch?v=YVCtaXw6fw8&ab_channel=Gwanghyun%28Bradley%29Kim**

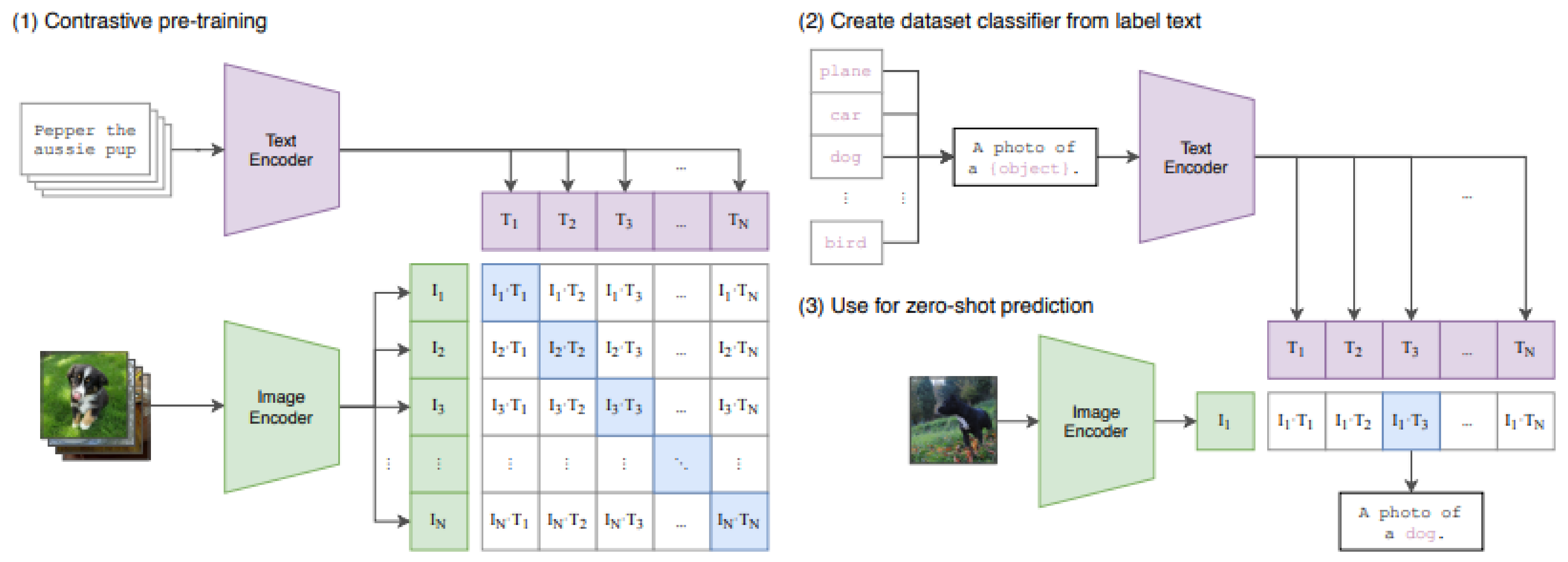
Motivation

- Successful manipulation of images should convert the image attribute to that of the target without unintended changes of the input content. Unfortunately, the current SOTA encoder-based GAN inversion approaches often fail to reconstruct images with novel poses, views, and details.

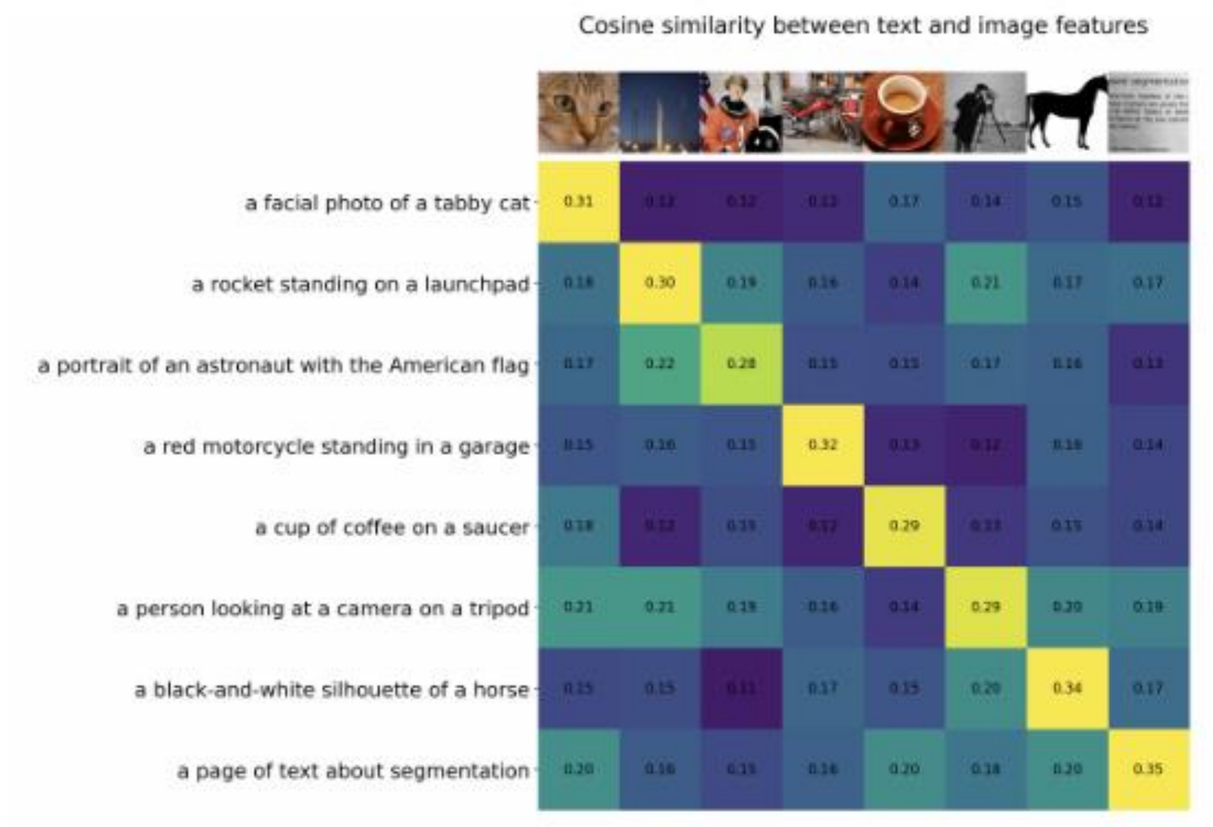


CLIP

- While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

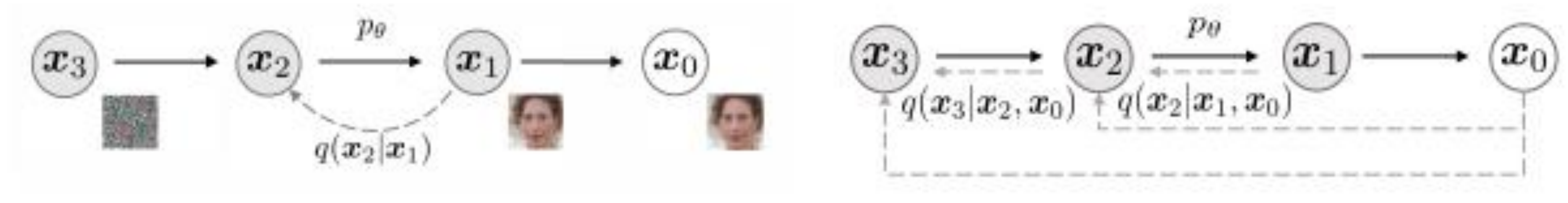


(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ... I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
I_1
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
A photo of a dog.

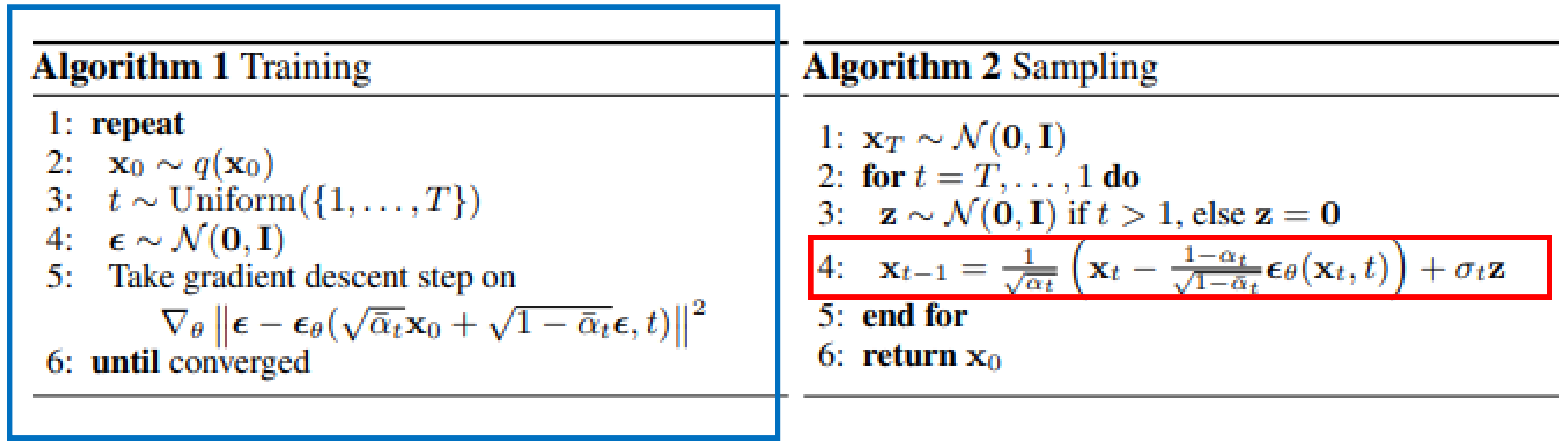


DDPM vs DDIM

- Diffusion (left) and non-Markovian (right) inference models.



Same as ddpm



Algorithm 1 Training

```
1: repeat
2:   $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3:   $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4:   $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5:   Take gradient descent step on
         $\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar\alpha_t}\mathbf{x}_0 + \sqrt{1-\bar\alpha_t}\boldsymbol{\epsilon}, t) \right\|^2$
6: until converged
```

Algorithm 2 Sampling

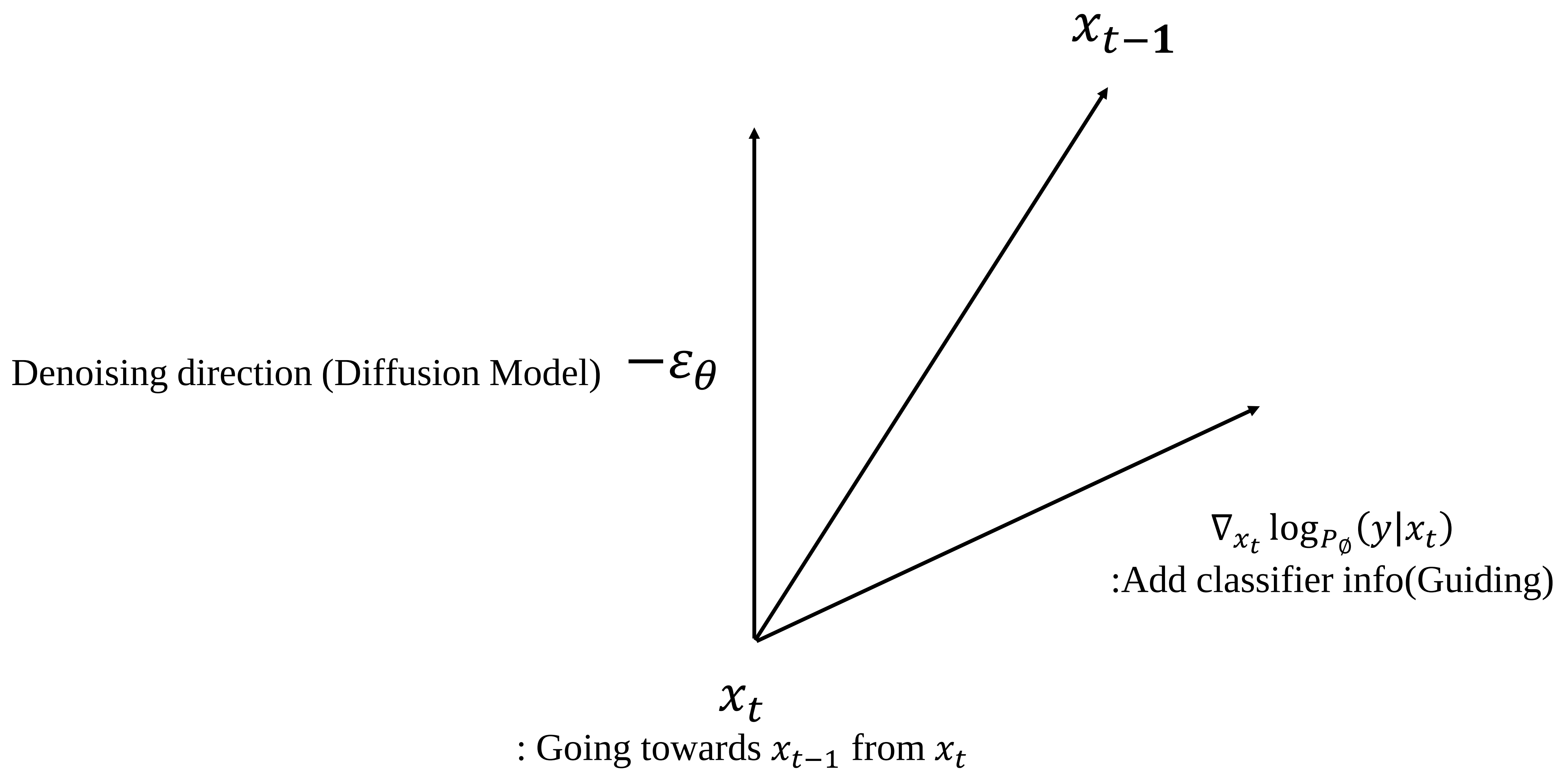
```
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: for $t = T, \ldots, 1$ do
3:   $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4:   $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar\alpha_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: end for
6: return $\mathbf{x}_0$
```

$$x_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\left(\frac{x_t - \sqrt{1-\alpha_t}\,\epsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}} \right)}_{\text{" predicted } x_0 \text{"}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_\theta^{(t)}(x_t)}_{\text{"direction pointing to } x_t \text{"}} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}} \longrightarrow$$

(eta)
ddpm = 1 -> stochastic
ddim = 0 -> deterministic

Classifier-free guidance

- Classifier guidance is a recently introduced method to trade off mode coverage and sample fidelity in conditional diffusion models post training, in the same spirit as low temperature sampling or truncation in other types of generative models.



x_{t-1}
Denoising direction (Diffusion Model) $-\varepsilon_\theta$
$\nabla_{x_t} \log_{P_\emptyset}(y|x_t)$
:Add classifier info(Guiding)
x_t
: Going towards x_{t-1} from x_t

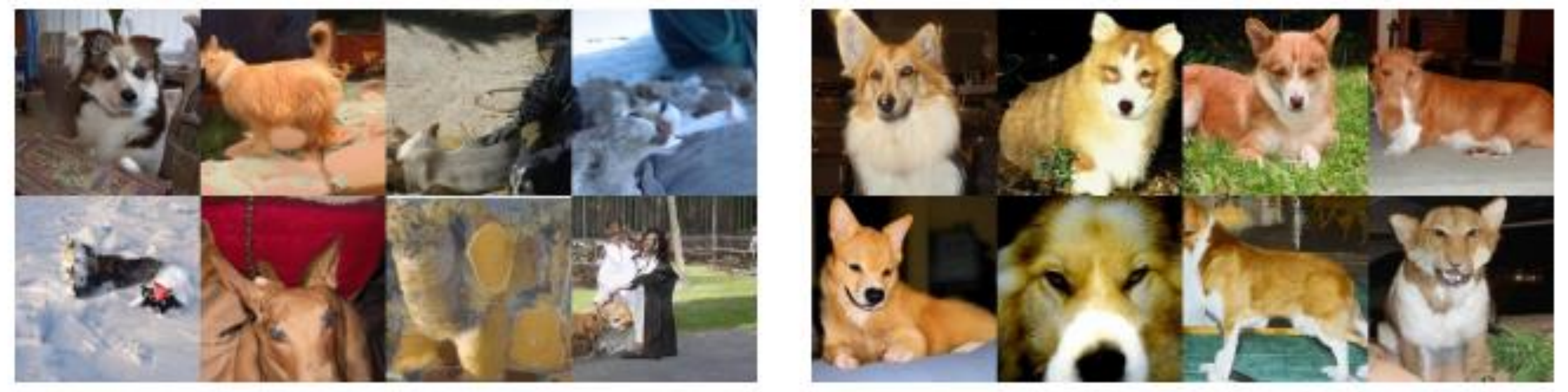


Figure 3: Samples from an unconditional diffusion model with classifier guidance to condition on the class "Pembroke Welsh corgi". Using classifier scale 1.0 (left; FID: 33.0) does not produce convincing samples in this class, whereas classifier scale 10.0 (right; FID: 12.0) produces much more class-consistent images.

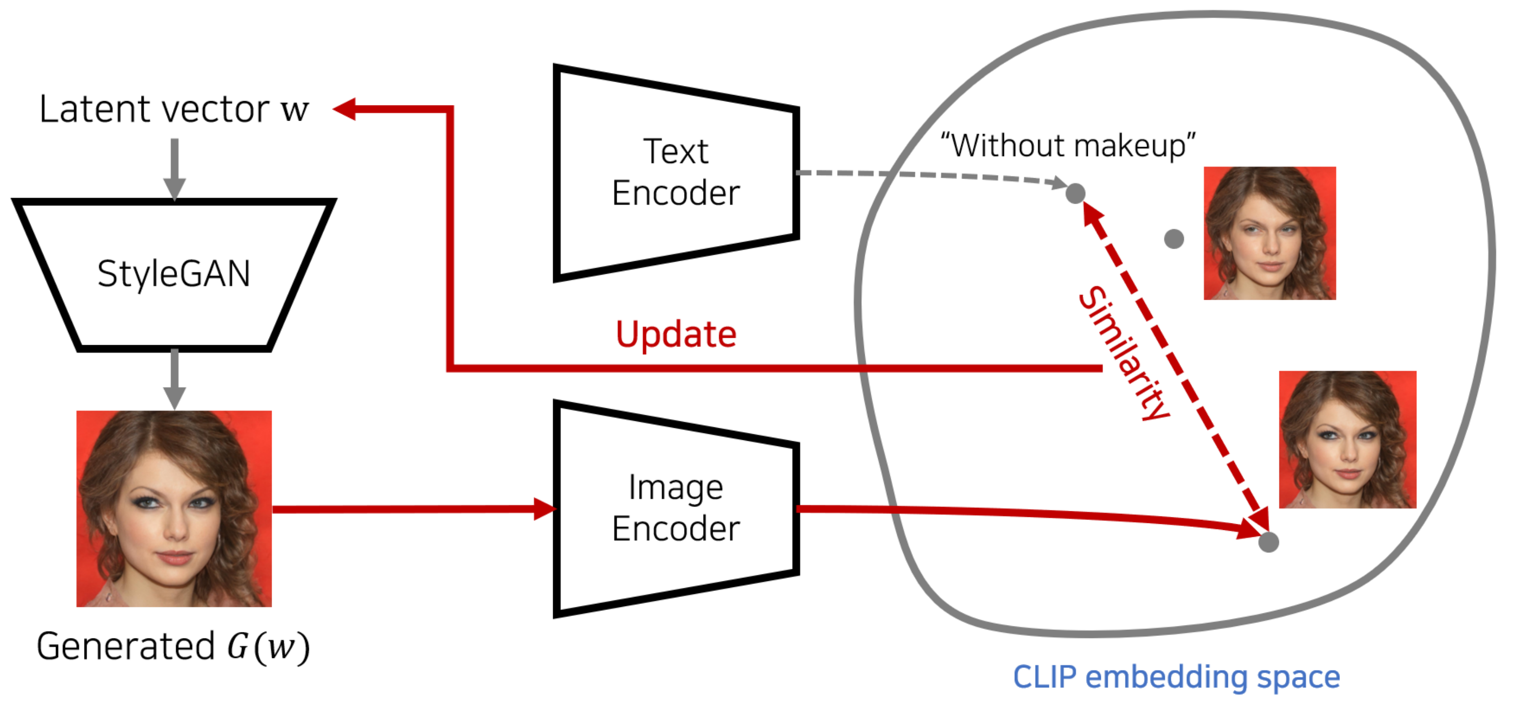
GAN Inversion-based + CLIP

- GAN: latent code(z) -> real image
- GAN inversion: real image -> latent code(z)

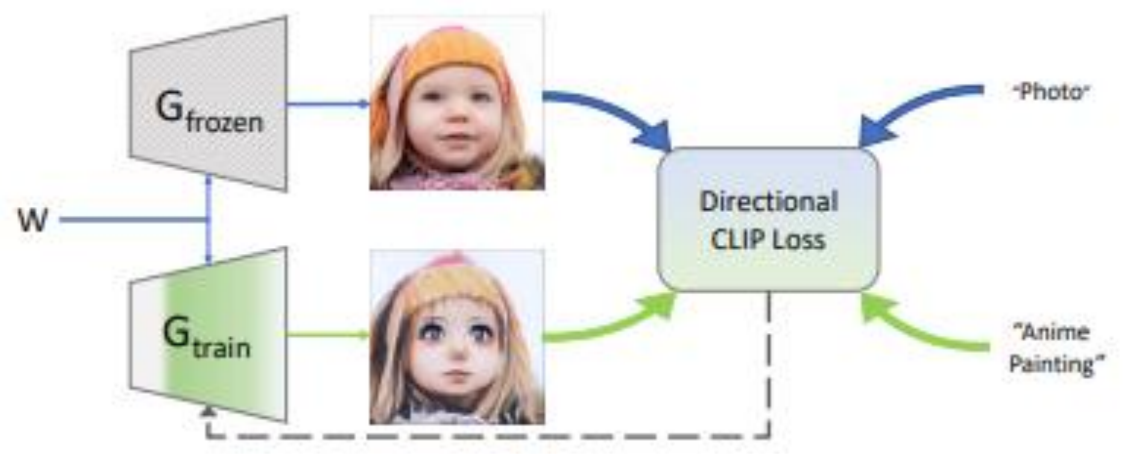
- Main goal: Aims to invert a given image back into the latent space of a pretrained GAN model. The image can then be faithfully reconstructed from the inverted code by the generator.

- Limitations: Hard to reconstruct novel detail & highly variable contents in image

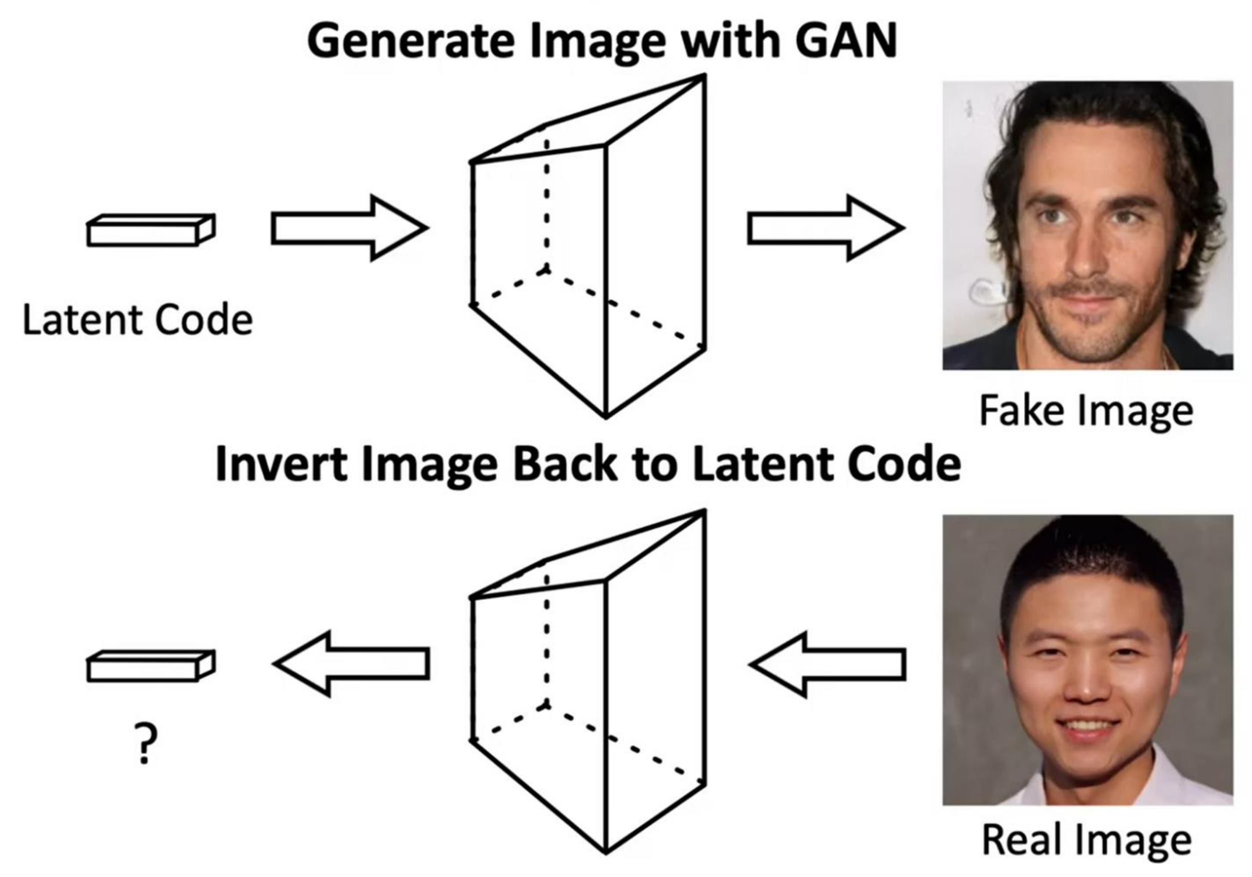
StyleCLIP



StyleGAN-NADA



GAN-Inversion

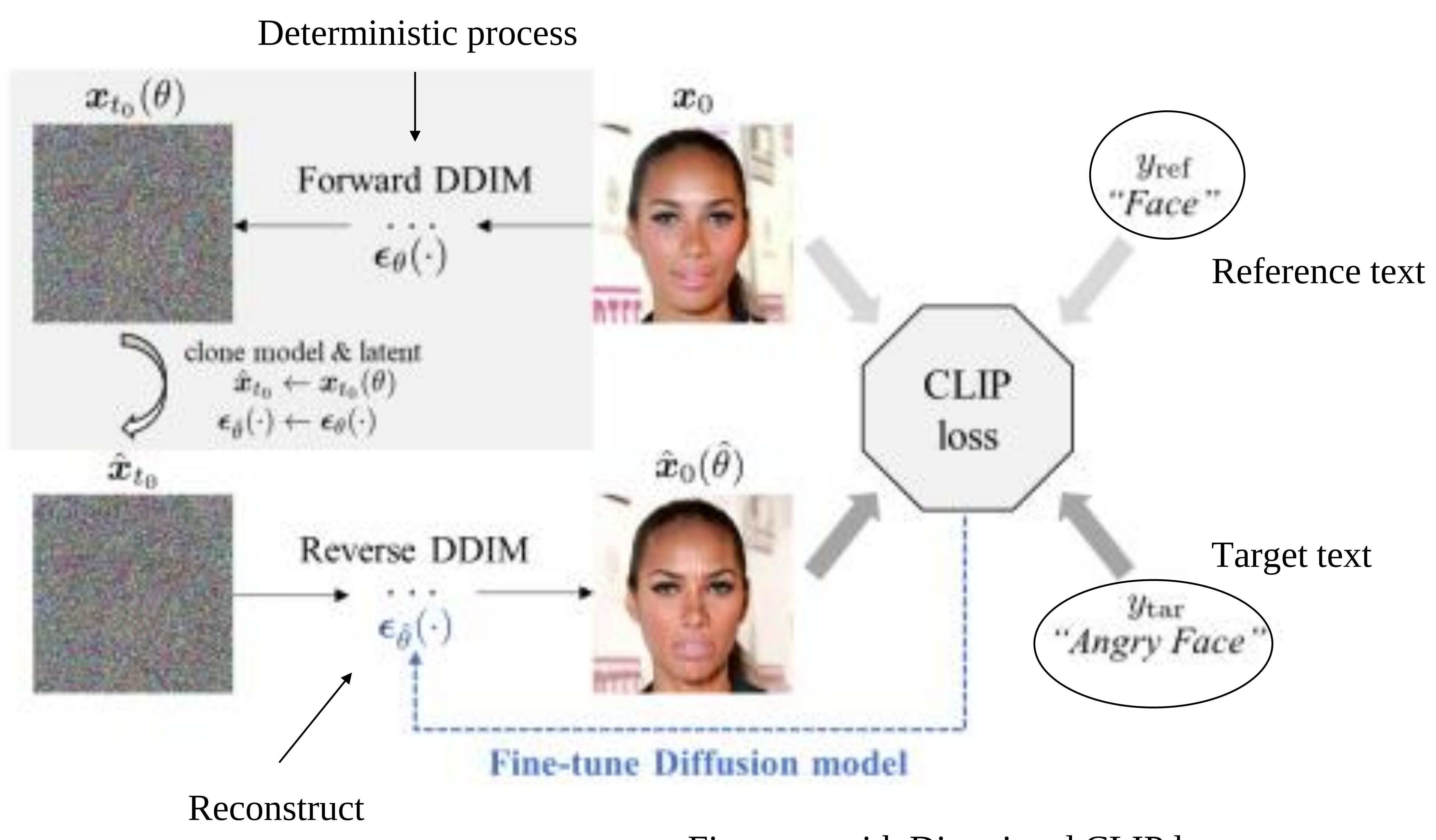


Overview

- GAN Inversion Method + CLIP enables zero-shot image manipulation guided by text prompts.
- However, their applications to diverse real images are still difficult due to the limited GAN inversion capability.
- Based on full inversion capability and high-quality image generation power of recent diffusion models, DiffusionCLIP performs zero shot image manipulation successfully.
- Unseen domains and takes another step towards general application by manipulating images from a widely varying ImageNet dataset. (General Application)
- Architecture improvement + classifier guidance

Overview + Loss

- Overview of DiffusionCLIP. The input image is first converted to the latent via diffusion models. Then, guided by directional CLIP loss, the diffusion model is fine-tuned, and the updated sample is generated during reverse diffusion.
- In the case of DDIM, the latent noises can be then inverted nearly perfectly to the original image using a reverse diffusion if the score function for the reverse diffusion is retained the same as that of the forward diffusion. Therefore the key idea of DiffusionCLIP is to fine-tune the score function in the reverse diffusion process using a CLIP loss that controls the attributes of the generated image based on the text prompts.



Fine tune with Directional CLIP loss

CLIP Loss :

$$\mathcal{L}_{\text{direction}}\left(\hat{x}_0(\hat\theta), y_{\text{tar}}; x_0, y_{\text{ref}}\right) + \underline{\mathcal{L}_{\text{id}}(\hat{x}_0(\hat\theta), x_0)},$$

case for face

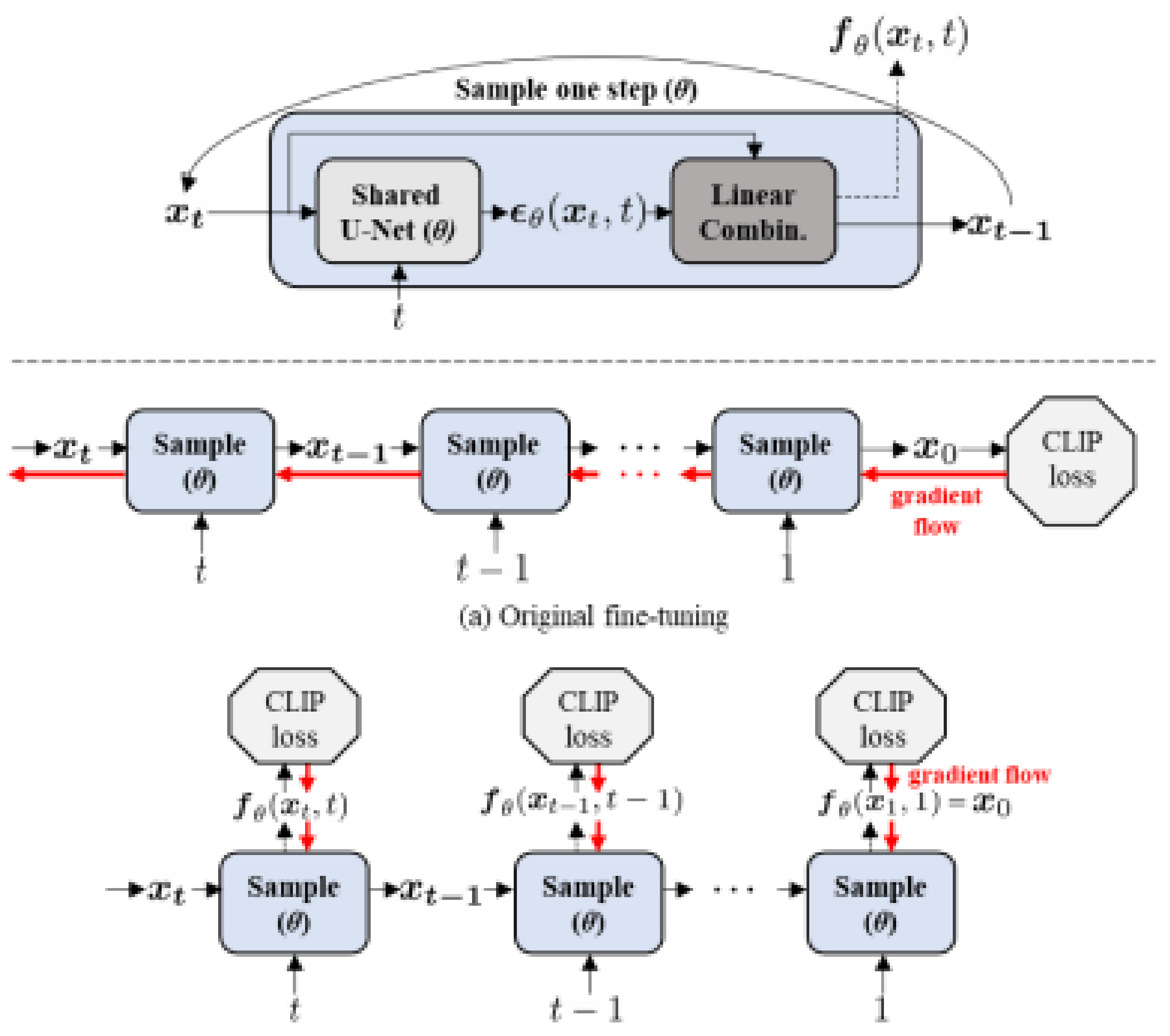
case for object

$$\mathcal{L}_{\text{id}}(\hat{x}_0(\hat\theta), x_0) = \underline{\lambda_{\text{L1}} \| x_0 - \hat{x}_0(\hat\theta) \|} + \underline{\lambda_{\text{face}} \mathcal{L}_{\text{face}}(\hat{x}_0(\hat\theta), x_0)},$$

Identity Loss: To prevent the unwanted changes and preserve the identity of the object. (L1 loss)

DiffusionCLIP Fine-tuning

- Gradient flows during fine-tunning the diffusion model with the shared architecture across t.
- Gradient flows during the fine-tunning and GPU-efficient fine-tunning. In GPU-efficient fine-tunning, loss calculation and a gradient step are proceeded at each time step t with $f_{\theta}(x_t - t)$ the prediction of x_0 and t.



(a) Original fine-tuning

(b) GPU-efficient fine-tuning

Algorithm 1: DiffuisonCLIP fine-tuning

```
Input: $\epsilon_\theta$ (pretrained model), $\{x_0^{(i)}\}_{i=1}^N$ (images to
       precompute), $y_{\mathrm{ref}}$ (reference text), $y_{\mathrm{tar}}$ (target
       text), $t_0$ (return step), $S_{\mathrm{for}}$ (# of inversion
       steps), $S_{\mathrm{gen}}$(# of generation steps), $K$ (# of
       fine-tuning iterations)
Output: $\epsilon_{\hat\theta}$ (fine-tuned model)
// Step 1: Precompute latents
1  Define $\{\tau_s\}_{s=1}^{S_{\mathrm{for}}}$ s.t. $\tau_1 = 0, \tau_{S_{\mathrm{for}}} = t_0$.
2  for $i - 1, 2, \dots, N$ do
3      for $s = 1, 2, \dots, S_{\mathrm{for}} - 1$ do
4          $\epsilon \leftarrow \epsilon_\theta(x_{\tau_s}^{(i)}, \tau_s)$;  $f \leftarrow f_\theta(x_{\tau_s}^{(i)}, \tau_s)$
5          $x_{\tau_{s+1}}^{(i)} \leftarrow \sqrt{\alpha_{\tau_{s+1}}} f + \sqrt{1 - \alpha_{\tau_{s+1}}} \epsilon$
6      Save the latent $x_{t_0}^{(i)} = x_{\tau_{S_{\mathrm{for}}}}^{(i)}$.
// Step 2: Update the diffusion model
7  Clone the pretrained model $\epsilon_{\hat\theta} \leftarrow \epsilon_\theta$
8  Define $\{\tau_s\}_{s=1}^{S_{\mathrm{gen}}}$ s.t. $\tau_1 = 0, \tau_{S_{\mathrm{gen}}} = t_0$.
9  for $k = 1, 2, \dots, K$ do
10     for $i = 1, 2, \dots, N$ do
11         Clone the latent $\hat{x}_{t_0}^{(i)} \leftarrow x_{t_0}^{(i)}$.
12         for $s = S_{\mathrm{gen}}, S_{\mathrm{gen}} - 1, \dots, 2$ do
13             $\epsilon \leftarrow \epsilon_{\hat\theta}(\hat{x}_{\tau_s}^{(i)}, \tau_s)$;  $f \leftarrow f_{\hat\theta}(\hat{x}_{\tau_s}^{(i)}, \tau_s)$
14             $\hat{x}_{\tau_{s-1}}^{(i)} \leftarrow \sqrt{\alpha_{\tau_{s-1}}} f + \sqrt{1 - \alpha_{\tau_{s-1}}} \epsilon$
15         $\mathcal{L}_{\mathrm{total}} \leftarrow \mathcal{L}_{\mathrm{direction}}(\hat{x}_0^{(i)}, y_{\mathrm{tar}}; x_0^{(i)}, y_{\mathrm{ref}})$
16         $\mathcal{L}_{\mathrm{total}} \leftarrow \mathcal{L}_{\mathrm{total}} + \mathcal{L}_{\mathrm{id}}(\hat{x}_0^{(i)}, x_0^{(i)})$
17         Take a gradient step on $\nabla_{\hat\theta} \mathcal{L}_{\mathrm{total}}$.
```

Algorithm 2: GPU-efficient fine-tuning

```
// Step 2: Update the diffusion model
1  Clone the pretrained model $\epsilon_{\hat\theta} \leftarrow \epsilon_\theta$
2  Define $\{\tau_s\}_{s=1}^{S_{\mathrm{gen}}}$ s.t. $\tau_1 = 0, \tau_{S_{\mathrm{gen}}} = t_0$.
3  for $k = 1, 2, \dots, K$ do
4      for $i = 1, 2, \dots, N$ do
5          Clone the latent $\hat{x}_{t_0}^{(i)} \leftarrow x_{t_0}^{(i)}$.
6          for $s = S_{\mathrm{gen}}, S_{\mathrm{gen}} - 1, \dots, 2$ do
7              $\epsilon \leftarrow \epsilon_{\hat\theta}(\hat{x}_{\tau_s}^{(i)}, \tau_s)$;  $f \leftarrow f_{\hat\theta}(\hat{x}_{\tau_s}^{(i)}, \tau_s)$
8              $\hat{x}_{\tau_{s-1}}^{(i)} \leftarrow \sqrt{\alpha_{\tau_{s-1}}} f + \sqrt{1 - \alpha_{\tau_{s-1}}} \epsilon$
9              $\mathcal{L}_{\mathrm{total}} \leftarrow \mathcal{L}_{\mathrm{direction}}(f, y_{\mathrm{tar}}; x_0^{(i)}, y_{\mathrm{ref}})$
10             $\mathcal{L}_{\mathrm{total}} \leftarrow \mathcal{L}_{\mathrm{total}} + \mathcal{L}_{\mathrm{id}}(f, x_0^{(i)})$
11             Take a gradient step on $\nabla_{\hat\theta} \mathcal{L}_{\mathrm{total}}$.
```

Novel Sampling Strategies

- Proposed a systematic approach to find the optimal sampling conditions that lead to high quality and speedy image manipulation.
- Return Step: Performs Diffusion up to t_0<T
- Discretization Step: [1, t_0], denoted as s_{for} and s_{gen} for forward diffusion and generative process
- When T=1000, the choices of $t_0 \in$ [300,600] and (s_{for}, s_{gen}) = (40,6) satisfy our goal. With these settings, the fine-tunning is finished in 1 ~ 7 minutes on NVIDIA Quardro RTX 6000

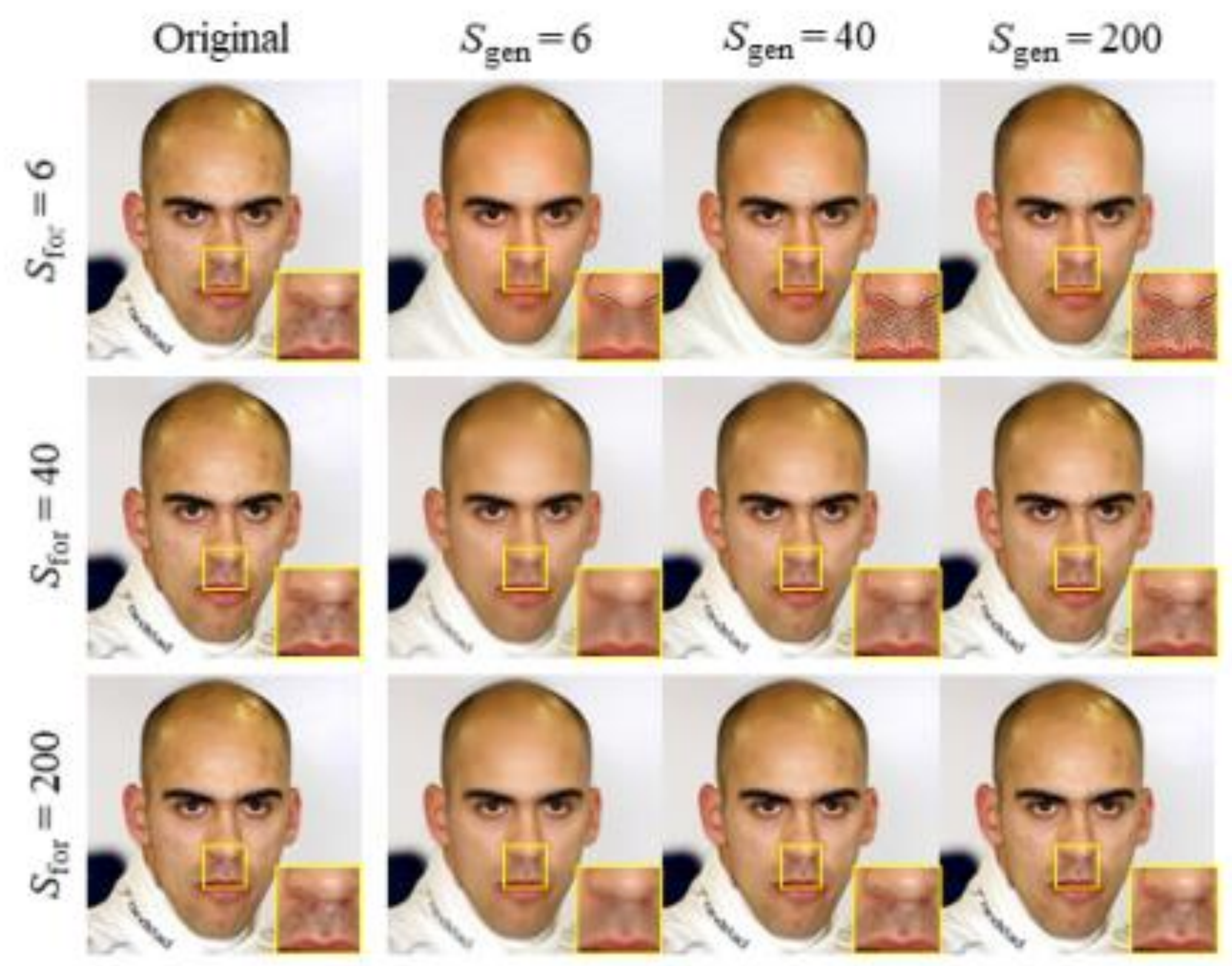


Figure 10. Reconstruction results varying the number of forward diffusion steps S_{for} and generative steps S_{gen}.

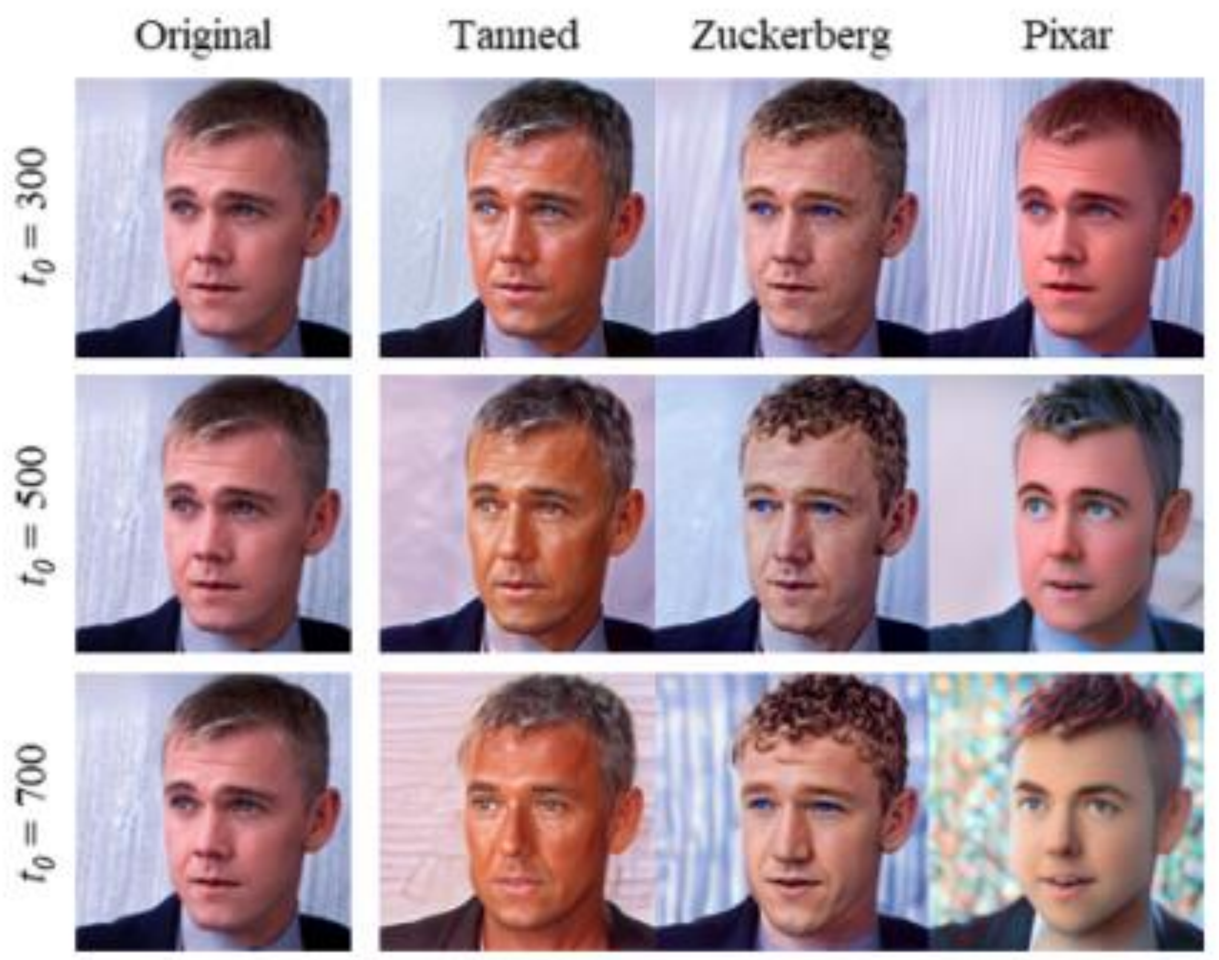
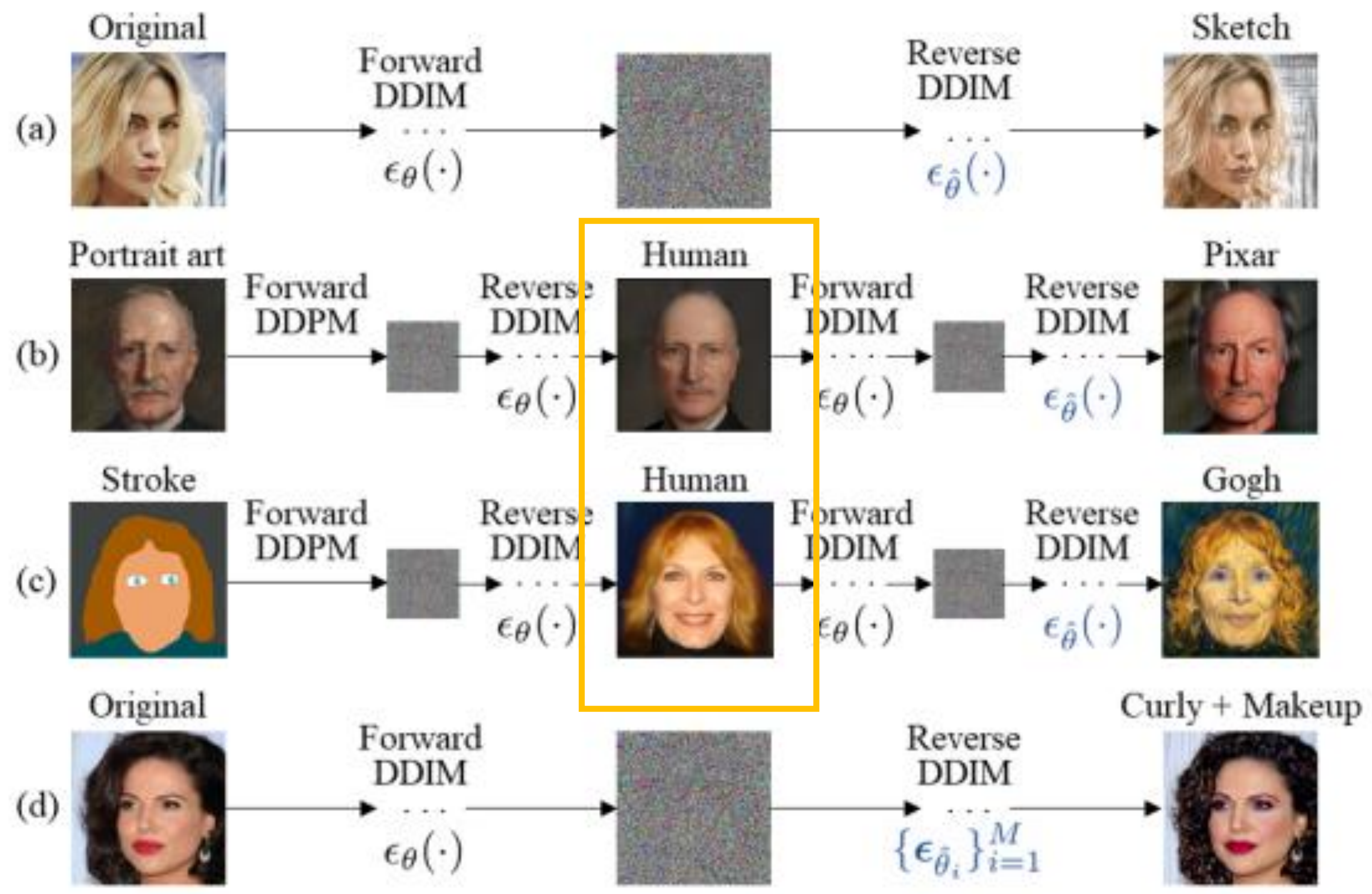


Figure 11. Manipulation results depending on t_0 values.

Novel Applications of DiffusionCLIP

(a) Manipulation of images in pretrained domain to CLIP-guided domain.
(b) Image translation between unseen domains
(c) Stroke-conditioned image generation in an unseen domain
(d) Multi-attribute transfer.



Experiment

Table 1. Quantitative comparison for face image reconstruction.

Method	MAE ↓	LPIPS ↓	SSIM ↑
Optimization	0.061	0.126	0.875
pSp	0.079	0.169	0.793
e4e	0.092	0.221	0.742
ReStyle w pSp	0.073	0.145	0.823
ReStyle w e4e	0.089	0.202	0.758
HFGI w e4e	0.062	0.127	0.877
Diffusion ($t_0 = 300$)	**0.020**	**0.073**	**0.914**
Diffusion ($t_0 = 400$)	0.021	0.076	0.910
Diffusion ($t_0 = 500$)	0.022	0.082	0.901
Diffusion ($t_0 = 600$)	0.024	0.087	0.893

Table 2. Human evaluation results of real image manipulation on CelebA-HQ [27]. The reported values mean the preference rate of results from DiffusionCLIP against each method.

vs		StyleGAN-NADA (+ Restyle w pSp)	StyleCLIP (+ e4e)
Hard cases	In-domain	69.85%	69.65%
	Out-of-domain	79.60%	94.60%
	All domains	73.10%	77.97%
General cases	In-domain	58.05%	50.10%
	Out-of-domain	71.03%	88.90%
	All domains	62.47%	63.03%

Table 3. Quantitative evaluation results. Our goal is to achieve the better score in terms of Directional CLIP similarity ($\mathcal{S}_{dir}$), segmentation-consistency (SC), and face identity similarity (ID).

	CelebA-HQ			LSUN-Church	
	$\mathcal{S}_{dir}$↑	SC↑	ID↑	$\mathcal{S}_{dir}$↑	SC↑
StyleCLIP	0.13	86.8%	0.35	0.13	67.9%
StyleGAN-NADA	0.16	89.4%	0.42	0.15	73.2%
DiffusionCLIP (Ours)	**0.17**	**93.7%**	**0.70**	**0.20**	**78.1%**

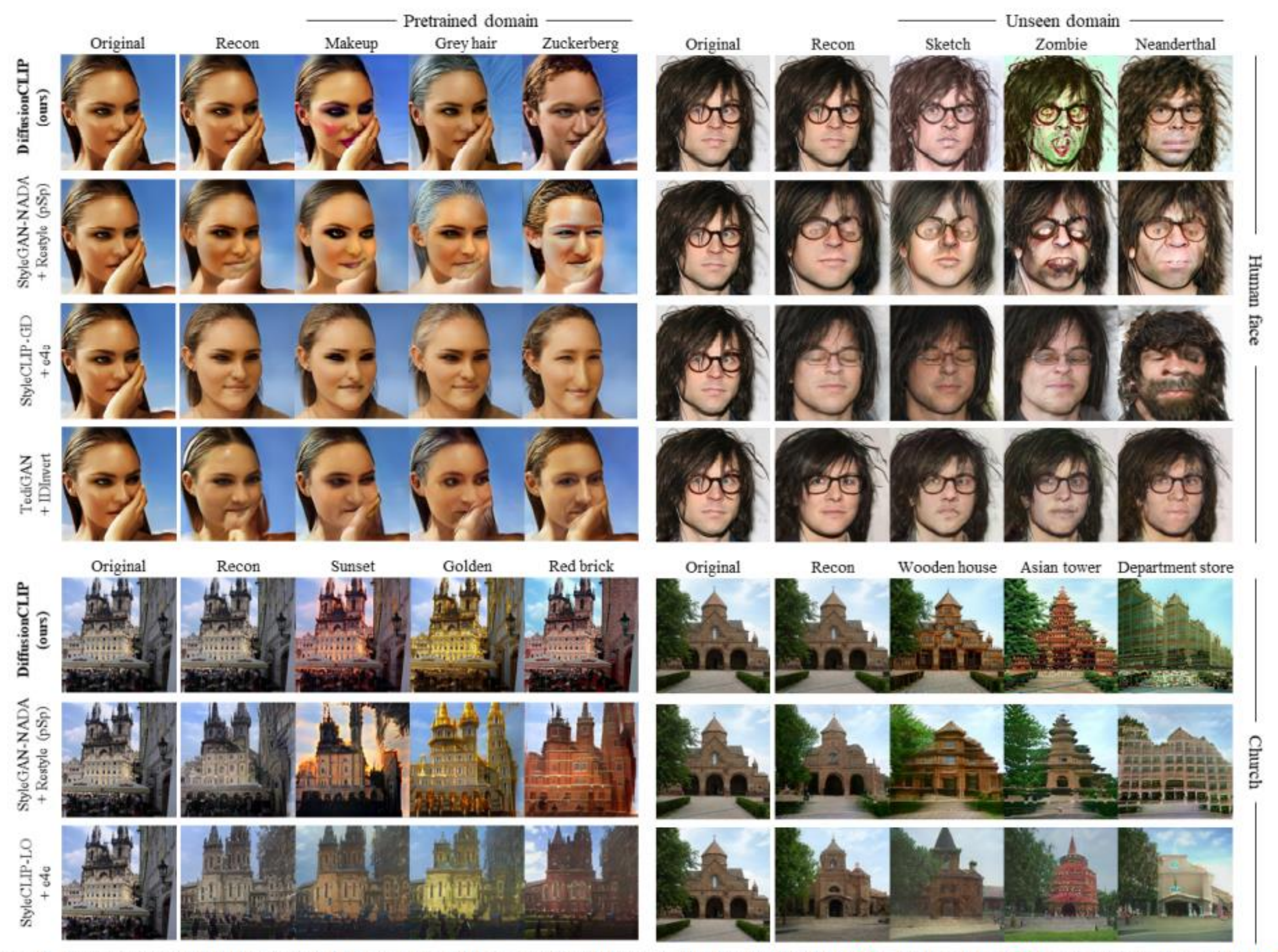


Figure 5. Comparison with the state-of-the-art text-driven manipulation methods: TediGAN [62], StyleCLIP [39] and StyleGAN-NADA [20]. StyleCLIP-LO and StyleCLIP-GD refer to the latent optimization (LO) and global direction (GD) methods of StyleCLIP.

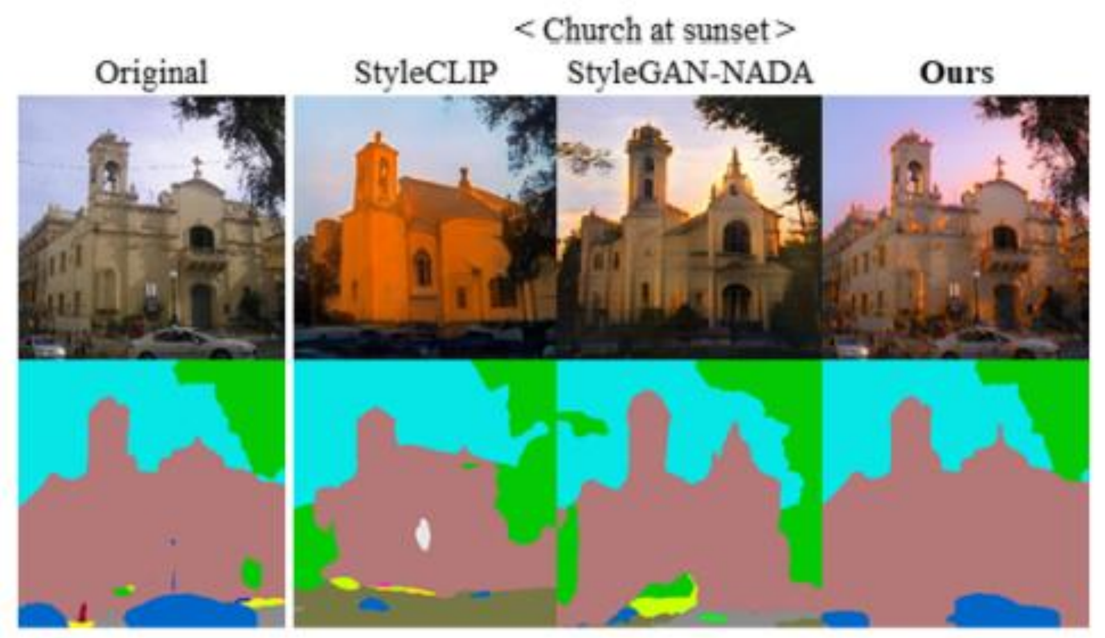


Figure 16. Example of segmentation results from the manipulation results by different methods.