ECCV 2024

Diffusion-Based Image-to-Image Translation by Noise Correction via Prompt Interpolation

Junsung Lee[1], Minsoo Kang[2], Bohyung Han[1,2]

[1]ECE & [2]IPAI, Seoul National University

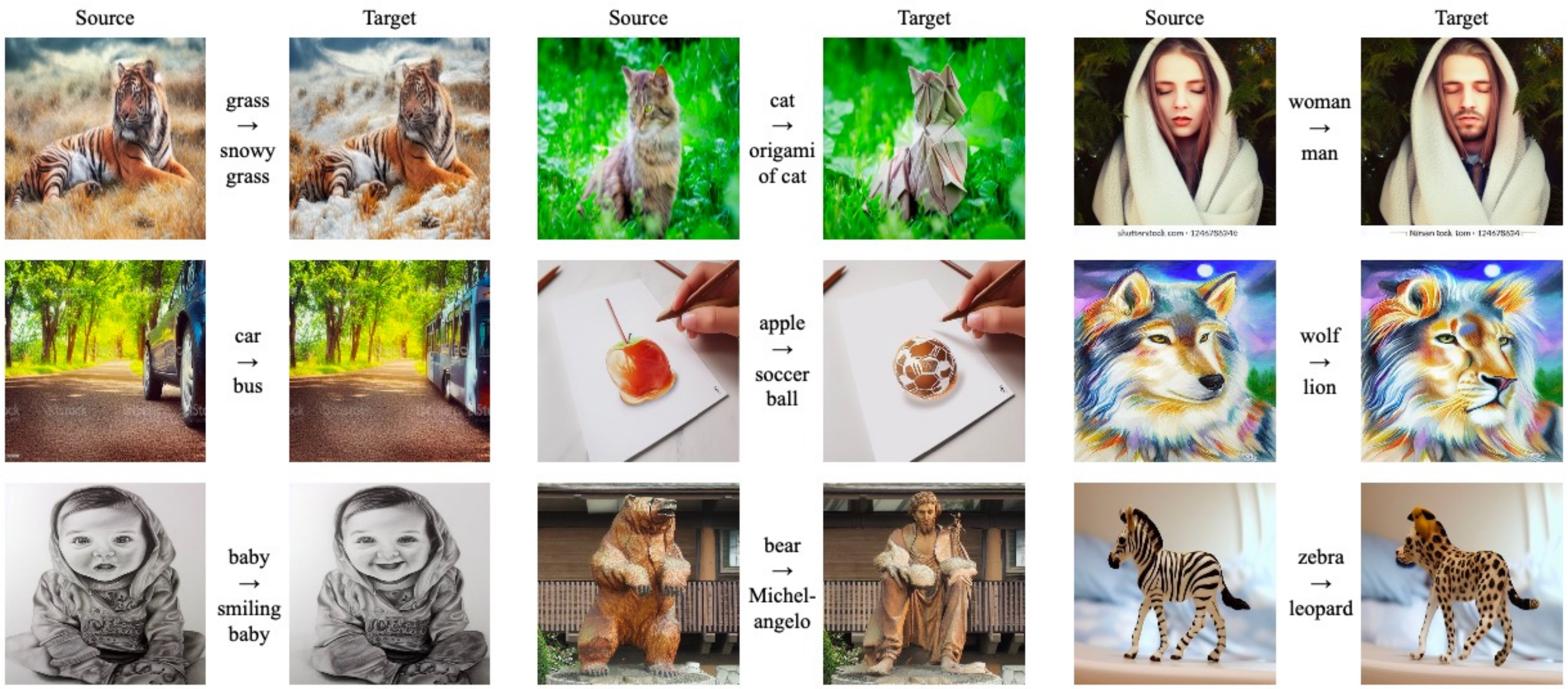
ComputerVisionLab
Seoul National University

EUROPEAN CONFERENCE ON COMPUTER VISION
MILANO 2024

Background

Text-Driven Image-to-Image(I2I) Translation

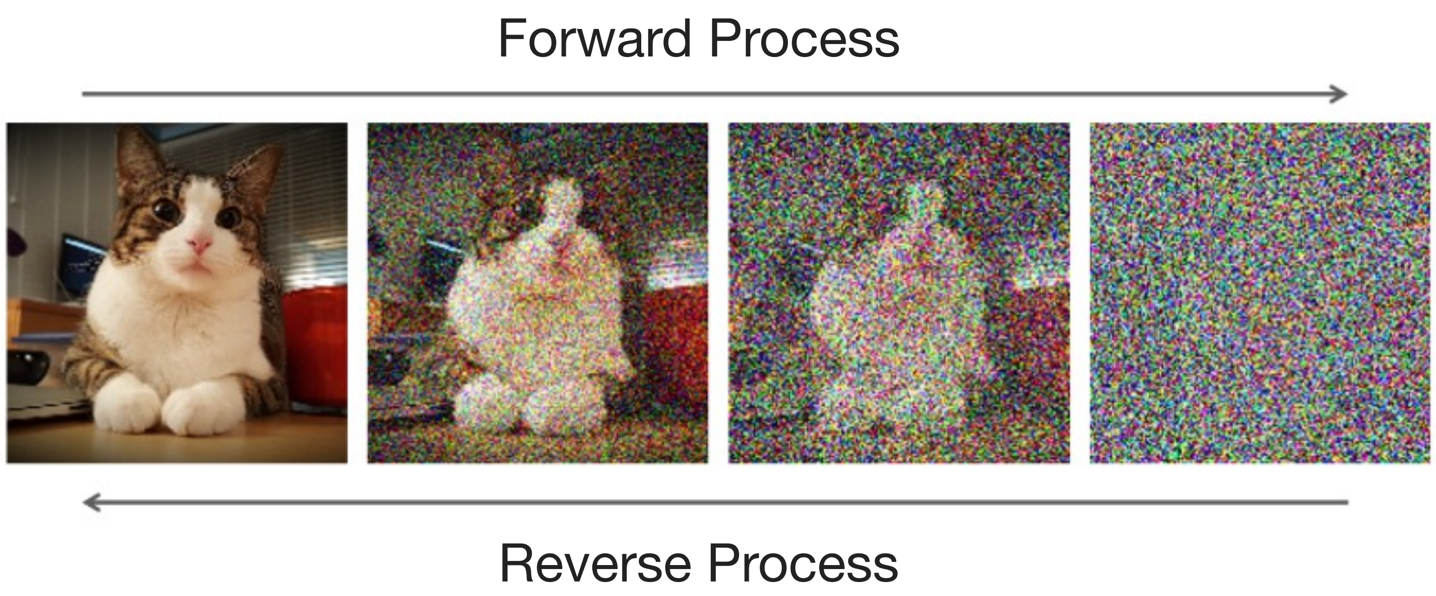
- Transforming input images into other images aligned with target prompts
- Diffusion-based Method



Background

Diffusion Process

- Forward Process: Input Images → Gaussian noises (Noising)
- Reverse Process: Gaussian noises → Generated Images (Denoising)
- Shared U-Net



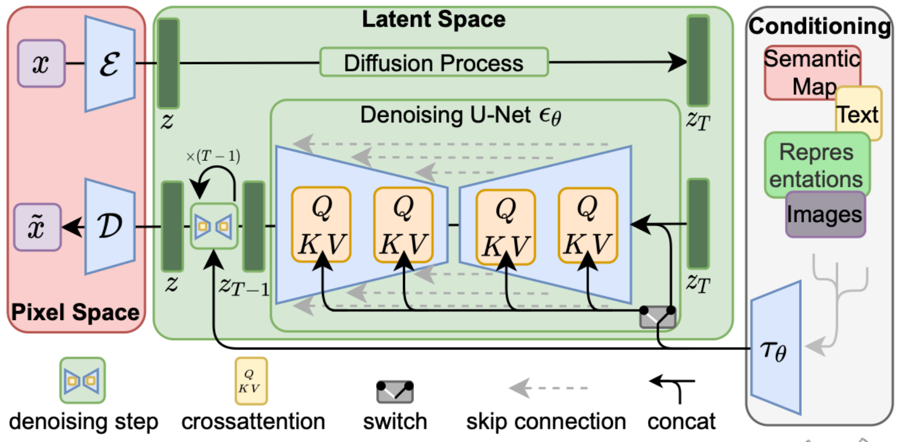


Image source: https://pub.towardsai.net/gan-is-diffusion-all-you-need-5ef127fa4ca / https://arxiv.org/pdf/2112.10752

Background

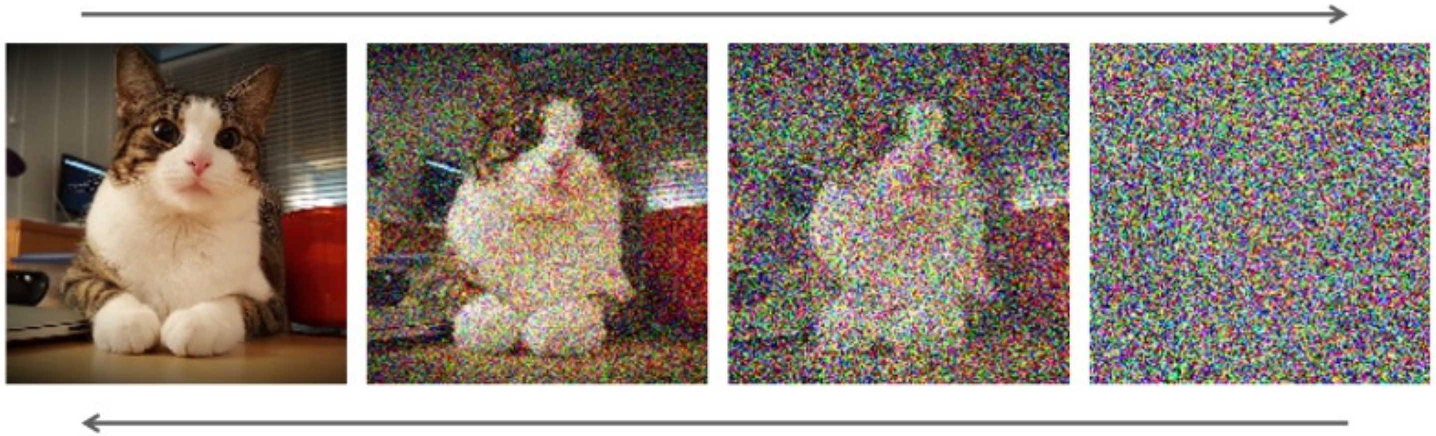
- Derived as $f_\theta(\mathbf{x}_t, t, \mathbf{y}) = \dfrac{\mathbf{x}_t - \sqrt{1-\alpha_t}\,\epsilon_\theta(\mathbf{x}_t, t, \mathbf{y})}{\sqrt{\alpha_t}}$
- Forward Process

$$\mathbf{x}_{t+1}^{\mathrm{src}} = \sqrt{\alpha_{t+1}} f_\theta(\mathbf{x}_t^{\mathrm{src}}, t, \mathbf{y}^{\mathrm{src}}) + \sqrt{1-\alpha_{t+1}}\,\epsilon_\theta(\mathbf{x}_t^{\mathrm{src}}, t, \mathbf{y}^{\mathrm{src}})$$

- Reverse Process

$$\mathbf{x}_{t-1}^{\mathrm{tgt}} = \sqrt{\alpha_{t-1}} f_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{tgt}}) + \sqrt{1-\alpha_{t-1}}\,\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{tgt}})$$

Forward Process



Reverse Process

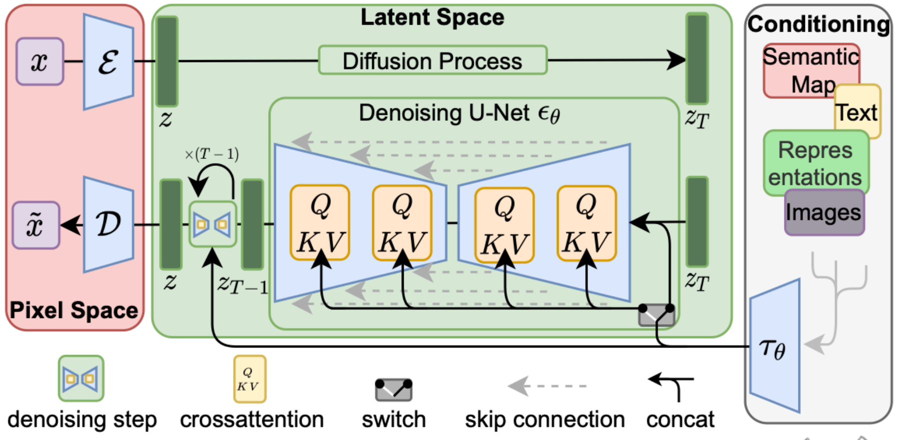


Image source: https://pub.towardsai.net/gan-is-diffusion-all-you-need-5ef127fa4ca / https://arxiv.org/pdf/2112.10752

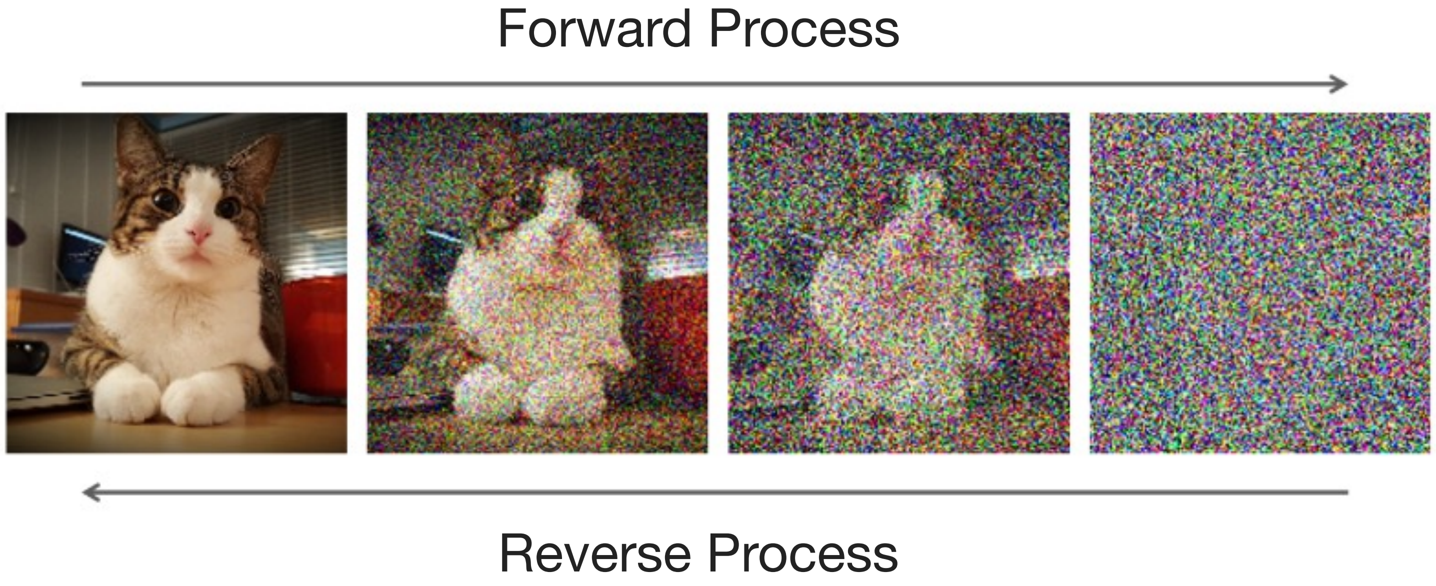
Background

- Problem

The starting point of the reverse process $\mathbf{x}_T^{\mathrm{tgt}}(=\mathbf{x}_T^{\mathrm{src}})$ is different from its true position $\mathbf{x}_T^{\mathrm{tgt}^*}$.

- Goal

Re-routing the reverse process without an additional training



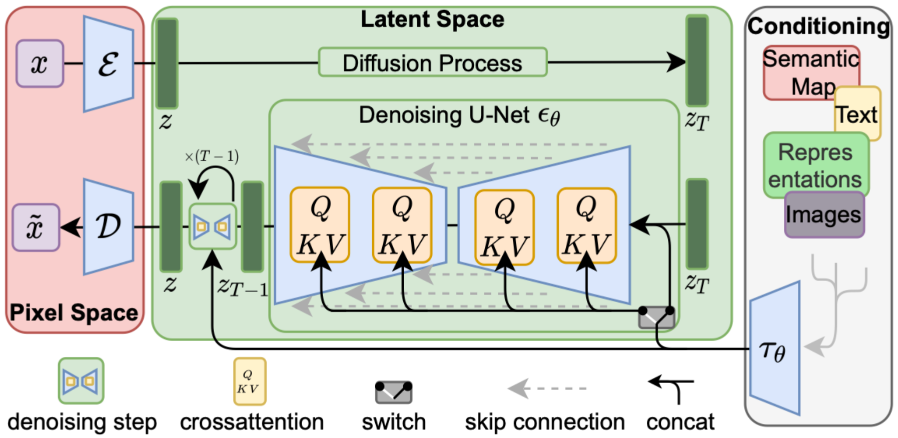


Image source: https://pub.towardsai.net/gan-is-diffusion-all-you-need-5ef127fa4ca / https://arxiv.org/pdf/2112.10752

Method

- Problem

The poor quality of I2I Translation when using naïve process

- Analysis

This occurs due to abrupt transition of text embeddings.

- Contribution

We introduce a correction term using prompt interpolation to smoothly edit source images.

Method: Correction Term

$$\hat{\epsilon}_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{tgt}}) := \epsilon_\theta(\mathbf{x}_t^{\mathrm{src}}, t, \mathbf{y}^{\mathrm{src}}) + \gamma \Delta\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}_t)$$

- First term: Source Noise

Preserving the structure / background of source images

- Second term: Correction Term

Desired as the noise to edit specific regions

- Goal: How to get ”Correction Term” without additional training?

Method: Correction Term

$$\hat{\epsilon}_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{tgt}}) := \epsilon_\theta(\mathbf{x}_t^{\mathrm{src}}, t, \mathbf{y}^{\mathrm{src}}) + \gamma \Delta\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}_t)$$

$$\Delta\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}_t) := \epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}_t) - \epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{src}})$$

- Correction Term

Difference between two noises conditioned by source embedding & interpolated embedding

- Interpolated Embedding $\mathbf{y}_t$

First steps: Similar to source embedding $\mathbf{y}^{\mathrm{src}}$

Final steps: Similar to target embedding $\mathbf{y}^{\mathrm{tgt}}$

Method: Prompt Interpolation

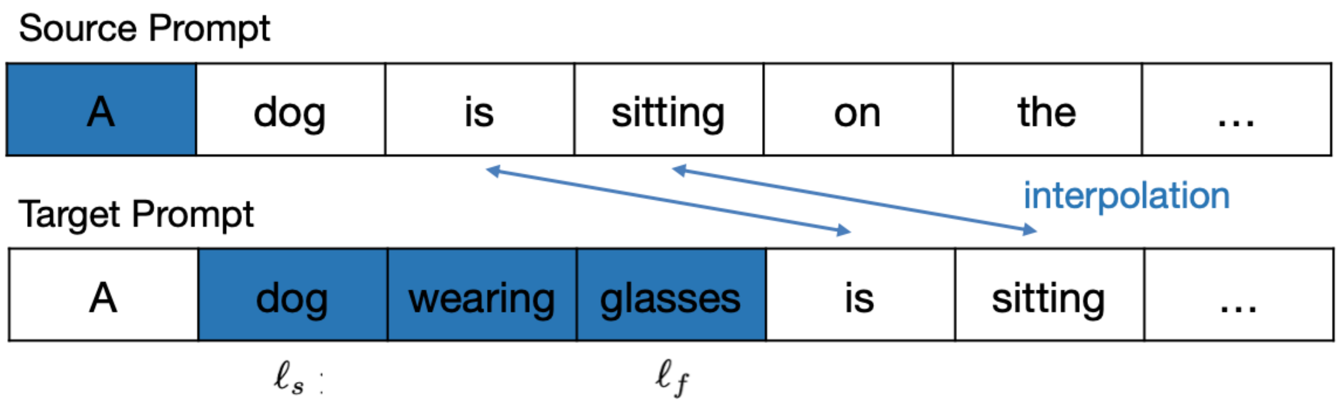
- Word Swap (dog → cat)

$$\mathbf{y}_t[\ell] = \beta_t \mathbf{y}^{\mathrm{tgt}}[\ell] + (1-\beta_t)\mathbf{y}^{\mathrm{src}}[\ell]$$

- Adding Phrases (dog → dog wearing glasses)

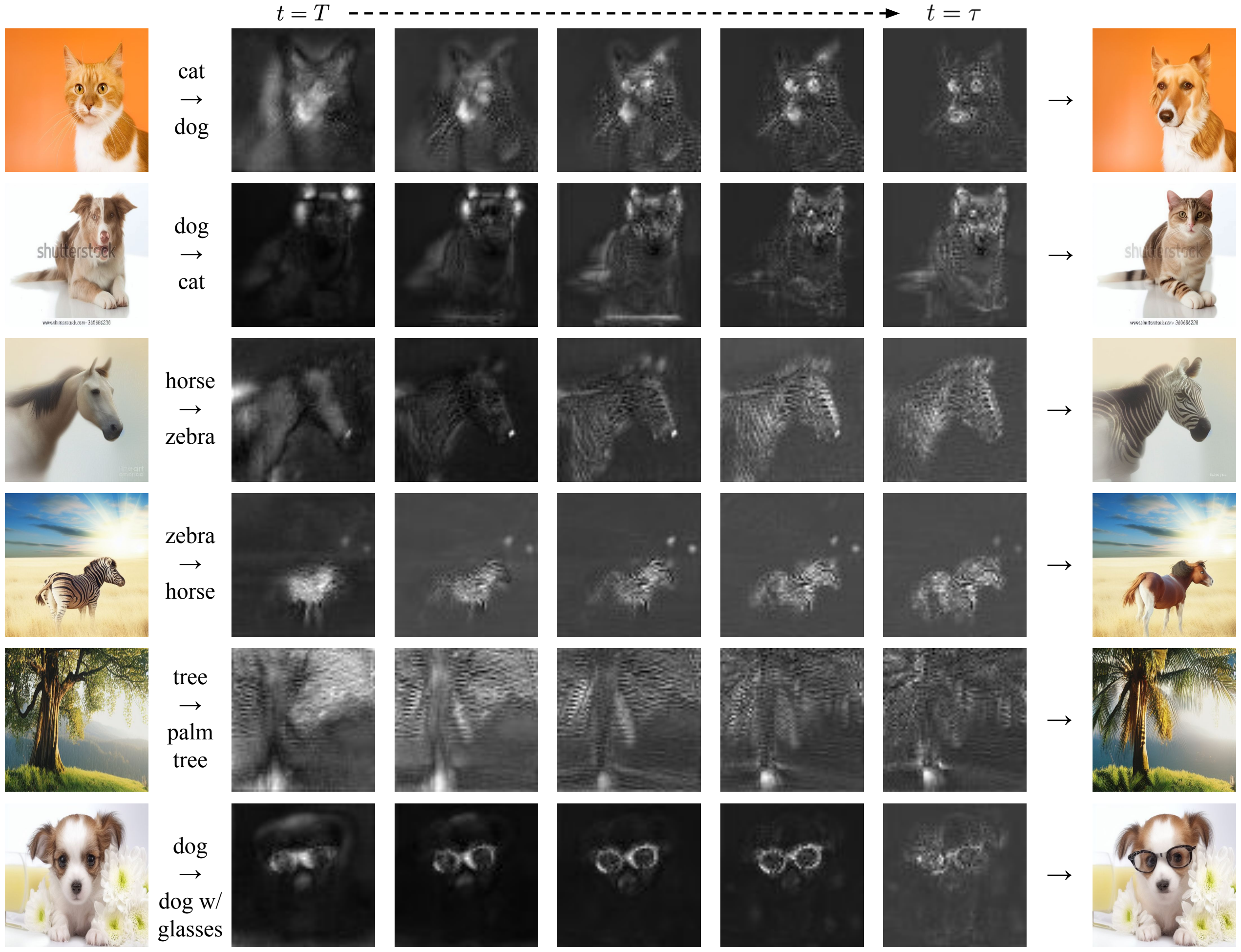
$$\mathbf{y}_t[\ell] = \begin{cases} \mathbf{y}^{\mathrm{src}}[\ell], & \text{if } \ell < \ell_s \\ \mathbf{y}^{\mathrm{tgt}}[\ell], & \text{if } \ell_s \leq \ell \leq \ell_f \\ \beta_t \mathbf{y}^{\mathrm{tgt}}[\ell] + (1-\beta_t)\mathbf{y}^{\mathrm{src}}[\ell - \ell_f + \ell_s], & \text{if } \ell > \ell_f \end{cases}$$

- Coefficient $\beta_t := \beta + (1-\beta) \times \dfrac{T-t}{T}$



Method: Prompt Interpolation

- Visualization of Correction Term



(a) Source (b) Noise Correction at time step t (c) Target

Method: PIC (Prompt Interpolation-based Correction)

- Pseudo Code

Algorithm 1 Target image generation by PIC

Input: source image $\mathbf{x}_0^{\mathrm{src}}$, source prompt embedding $\mathbf{y}^{\mathrm{src}}$, target prompt embedding $\mathbf{y}^{\mathrm{tgt}}$, hyperparameters β, γ, τ
for $t \leftarrow 0, \cdots, T-1$ **do**
    Compute $\epsilon_\theta(\mathbf{x}_t^{\mathrm{src}}, t, \mathbf{y}^{\mathrm{src}})$ and obtain $\mathbf{x}_{t+1}^{\mathrm{src}}$ by Eq. (1) while saving $\epsilon_\theta(\mathbf{x}_t^{\mathrm{src}}, t, \mathbf{y}^{\mathrm{src}})$
end for
$\mathbf{x}_T^{\mathrm{tgt}} \leftarrow \mathbf{x}_T^{\mathrm{src}}$
for $t \leftarrow T, \cdots, T-\tau+1$ **do**
    Obtain $\mathbf{y}_t$ based on $\mathbf{y}^{\mathrm{src}}$ and $\mathbf{y}^{\mathrm{tgt}}$ using Eq. (8) or Eq. (10) depending on the given task
    Compute $\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}_t)$ and $\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{src}})$
    Obtain the revised model $\hat{\epsilon}_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{tgt}})$ using Eq. (7)
    Obtain $\mathbf{x}_{t-1}^{\mathrm{tgt}}$ using Eq. (3) by replacing $\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{tgt}})$ with $\hat{\epsilon}_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{tgt}})$
end for
for $t \leftarrow T-\tau, \cdots, 1$ **do**
    Obtain $\mathbf{x}_{t-1}^{\mathrm{tgt}}$ using Eq. (3)
end for
Output: target image $\mathbf{x}_0^{\mathrm{tgt}}$

Experiments

- 250 images in LAION-5B Dataset for 6 tasks
- Stable Diffusion v1.4, 50 diffusion steps
- Hyperparameters: $\gamma = 1.0$, $\tau = 25$, $\beta = 0.3$ (word swap) & 0.8 (adding phrases)
- Comparison with other image translation models
  - Prompt-to-Prompt (PtP)
  - Plug-and-Play (PnP)
  - Pix2Pix-Zero (P2P)
- 3 metrics
  - CS (CLIP Similarity)
  - BD (Background Distance)
  - SD (Structure Distance)

Experiments

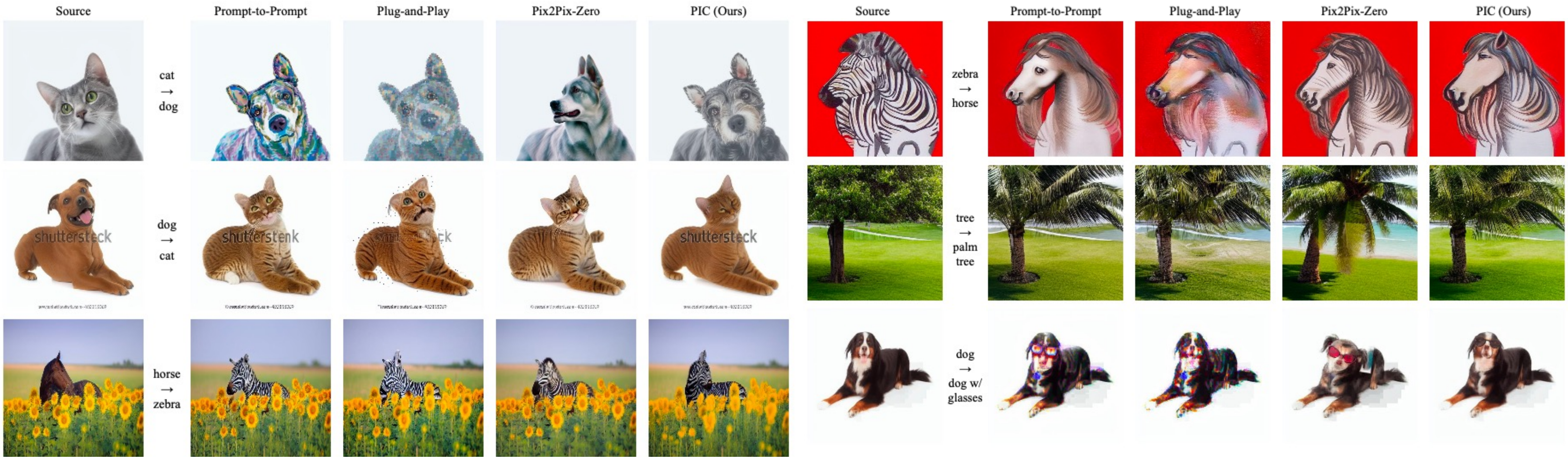
- Quantitative comparison with other algorithms

Black: Best Performance / Red: Second-best Performance

Task	PtP			PnP			P2P			PIC (Ours)		
	CS (↑)	BD (↓)	SD (↓)	CS (↑)	BD (↓)	SD (↓)	CS (↑)	BD (↓)	SD (↓)	CS (↑)	BD (↓)	SD (↓)
dog → cat	0.290	0.076	0.038	**0.293**	0.100	0.032	0.281	0.127	0.099	**0.293**	**0.045**	**0.031**
cat → dog	0.288	0.095	0.042	**0.291**	0.099	**0.033**	0.282	0.100	0.054	0.288	**0.057**	**0.033**
horse → zebra	0.320	0.133	0.042	**0.333**	0.158	0.042	0.323	0.193	0.078	0.324	**0.085**	**0.037**
zebra → horse	0.291	0.183	0.051	**0.299**	0.152	**0.043**	0.282	0.216	0.104	0.292	**0.126**	0.050
tree → palm tree	**0.315**	0.147	0.045	0.314	0.122	0.039	0.314	0.129	0.046	0.314	**0.085**	**0.036**
dog → dog w/glasses	0.310	0.041	0.020	0.302	0.087	0.025	**0.322**	0.050	**0.015**	0.312	**0.026**	0.016
Average	0.302	0.113	0.040	**0.305**	0.120	0.036	0.301	0.136	0.066	0.304	**0.071**	**0.034**

Experiments

- Qualitative comparison with other algorithms



Experiments

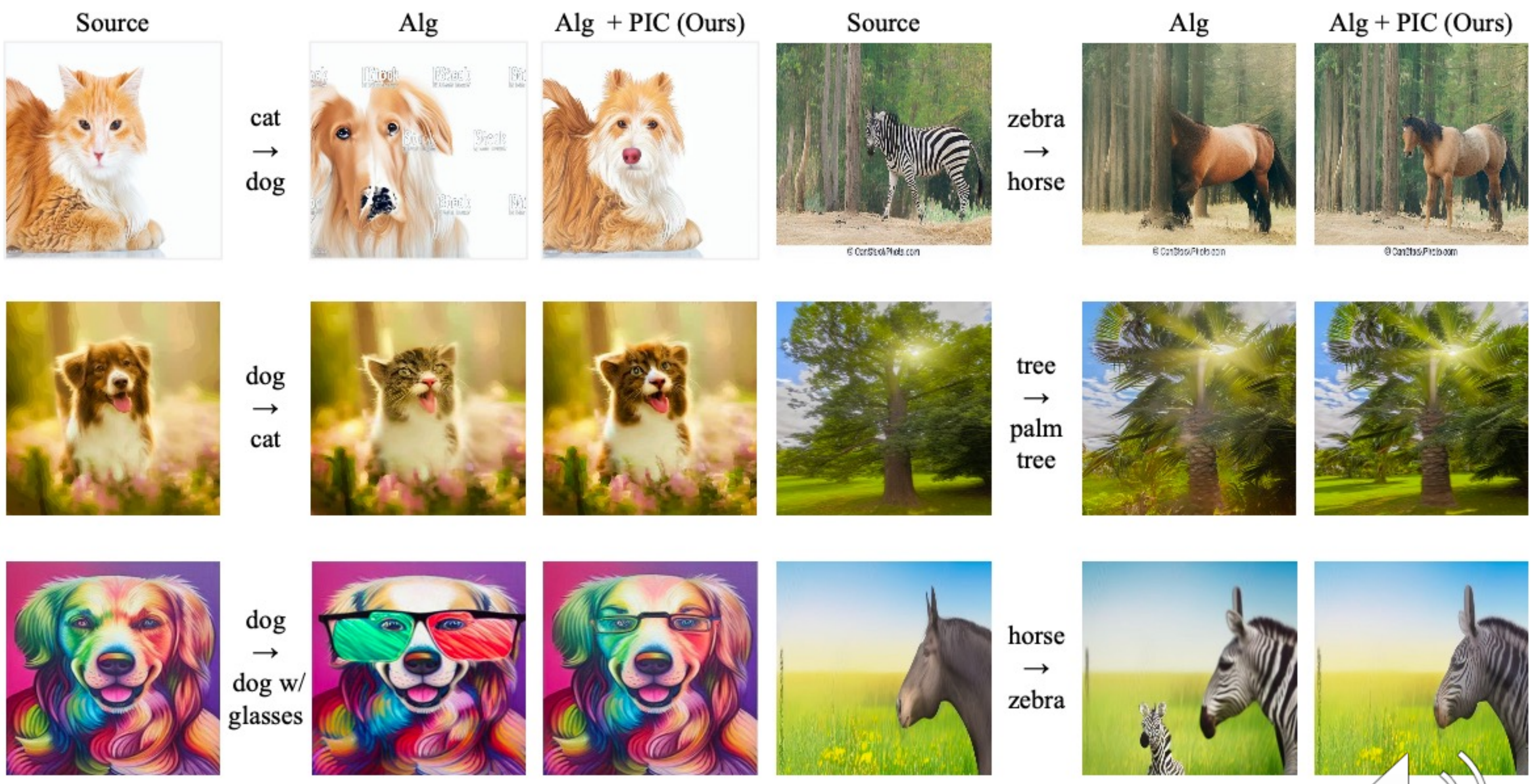
- Quantitative comparison: [Algorithms] vs [Algorithms] + PIC

Prompt-to-Prompt (PtP), Plug-and-Play (PnP), Pix2Pix-Zero (P2P)

Task	PtP			PtP + PIC (Ours)		
	CS (↑)	BD (↓)	SD (↓)	CS (↑)	BD (↓)	SD (↓)
dog → cat	**0.290**	0.076	0.038	0.283	**0.051**	**0.021**
cat → dog	0.288	0.095	0.042	**0.291**	**0.052**	**0.027**
horse → zebra	**0.320**	0.133	0.042	0.292	**0.071**	**0.018**
zebra → horse	**0.291**	0.183	0.051	0.290	**0.131**	**0.034**
tree → palm tree	**0.315**	0.147	0.045	0.301	**0.070**	**0.026**
dog → dog w/glasses	**0.310**	0.041	0.020	0.301	**0.038**	**0.011**
Average	**0.302**	0.113	0.040	0.295	**0.069**	**0.023**

Task	PnP			PnP + PIC (Ours)		
	CS (↑)	BD (↓)	SD (↓)	CS (↑)	BD (↓)	SD (↓)
dog → cat	**0.293**	0.100	0.032	0.282	**0.092**	**0.027**
cat → dog	**0.291**	0.099	0.033	0.288	**0.083**	**0.028**
horse → zebra	**0.333**	0.158	0.042	0.317	**0.121**	**0.035**
zebra → horse	**0.299**	0.152	0.043	0.285	**0.135**	**0.037**
tree → palm tree	**0.314**	0.122	0.039	0.295	**0.070**	**0.024**
dog → dog w/glasses	**0.302**	0.087	0.025	0.300	**0.085**	**0.024**
Average	**0.305**	0.120	0.036	0.295	**0.098**	**0.029**

Task	P2P			P2P + PIC (Ours)		
	CS (↑)	BD (↓)	SD (↓)	CS (↑)	BD (↓)	SD (↓)
dog → cat	0.281	0.127	0.099	**0.282**	**0.051**	**0.017**
cat → dog	0.282	0.100	0.054	**0.285**	**0.056**	**0.016**
horse → zebra	**0.323**	0.193	0.078	0.309	**0.070**	**0.016**
zebra → horse	**0.282**	0.216	0.104	0.279	**0.117**	**0.017**
tree → palm tree	**0.314**	0.129	0.046	0.298	**0.047**	**0.014**
dog → dog w/glasses	**0.322**	**0.050**	0.015	0.302	0.053	**0.011**
Average	**0.301**	0.136	0.066	0.293	**0.066**	**0.015**



Experiments

- Inference Time (Evaluation on A6000 GPU)

	PtP	PnP	P2P	PIC (Ours)
Inference time (s)	31.2	**24.4**	52.2	**18.1**

- Contribution of Noise Correction(NC) and Prompt Interpolation(PI)

Including both NC and PI achieves better performance than others.

Task	DDIM			DDIM+PI			DDIM+NC			PIC (Ours)		
	CS (↑)	BD (↓)	SD (↓)	CS (↑)	BD (↓)	SD (↓)	CS (↑)	BD (↓)	SD (↓)	CS (↑)	BD (↓)	SD (↓)
dog → cat	**0.289**	0.158	0.086	**0.289**	0.130	0.070	**0.293**	**0.054**	**0.038**	**0.293**	**0.045**	**0.031**
cat → dog	0.283	0.185	0.089	**0.285**	0.150	0.070	**0.288**	**0.068**	**0.041**	**0.288**	**0.057**	**0.033**
horse → zebra	0.325	0.287	0.123	**0.330**	0.214	0.097	**0.333**	**0.113**	**0.050**	0.324	**0.085**	**0.037**
zebra → horse	**0.294**	0.295	0.104	**0.294**	0.254	0.097	**0.294**	**0.139**	**0.055**	**0.292**	**0.126**	**0.050**
tree → palm tree	0.304	0.234	0.088	0.306	**0.222**	0.084	**0.312**	**0.085**	**0.056**	**0.314**	**0.085**	**0.036**
dog → dog w/glasses	**0.318**	0.134	0.072	0.310	0.132	0.065	**0.317**	**0.029**	**0.021**	0.312	**0.026**	**0.016**
Average	0.302	0.216	0.094	0.302	0.184	0.081	**0.306**	**0.081**	**0.044**	**0.304**	**0.071**	**0.034**

Thank you!

https://github.com/JS-Lee525/PIC

leejs0525@snu.ac.kr

ComputerVisionLab
Seoul National University