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

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

[1]ECE & [2]IPAI, Seoul National University
{leejs0525,kminsoo,bhhan}@snu.ac.kr

Abstract. We propose a simple but effective training-free approach tailored to diffusion-based image-to-image translation. Our approach revises the original noise prediction network of a pretrained diffusion model by introducing a noise correction term. We formulate the noise correction term as the difference between two noise predictions; one is computed from the denoising network with a progressive interpolation of the source and target prompt embeddings, while the other is the noise prediction with the source prompt embedding. The final noise prediction network is given by a linear combination of the standard denoising term and the noise correction term, where the former is designed to reconstruct must-be-preserved regions while the latter aims to effectively edit regions of interest relevant to the target prompt. Our approach can be easily incorporated into existing image-to-image translation methods based on diffusion models. Extensive experiments verify that the proposed technique achieves outstanding performance with low latency and consistently improves existing frameworks when combined with them.

Keywords: training-free image-to-image translation · diffusion models · generative modeling

1   Introduction

The diffusion probabilistic model [7,25–27] is currently a dominant framework for image generation. It has often been trained to generate high-fidelity images from text prompts [19,21,23], and has also been applied to image-to-image translation given a target text prompt [1, 5, 9–11, 13, 16, 29], where the goal is to modify local regions in a source image based on the target prompt while preserving its background or structure of the image. However, the text-driven image-to-image translation task is an inherently challenging problem, mainly because it is infeasible to find a desirable starting point of the reverse diffusion process for denoising and is difficult to exclusively edit specific regions of generated images without distorting the remaining parts.

   To tackle the critical challenges, several approaches rely on fine-tuning [1, 9, 10] for customizing pretrained diffusion-based denoising networks; they encourage the translated images to reflect the target prompt and preserve the

Fig. 1: Image-to-image translation results using the proposed method on data sampled from the LAION-5B dataset [24]. Our approach effectively preserves the structure and the background in source images while successfully editing the local region of interest.

background or the structure in the source image. On the other hand, training-free techniques [5, 11, 13, 16, 29] focus on manipulating denoising strategies used in the reverse process of diffusion models without incurring heavy training costs.

We present a simple but effective training-free image-to-image translation technique, which proposes a variation of the DDIM reverse process. Our approach estimates the noise correction term to generate desirable images relevant to target prompts, which is achieved by progressive prompt interpolation during the reverse process of diffusion models. The proposed noise prediction network for image-to-image translation is composed of two parts: (a) the denoising network output given the source latent and the source prompt and (b) a noise correction term defined as the difference between the two noise predictions of the target latent conditioned on the progressively interpolated embeddings and the source text embeddings. The first term ensures that the target image preserves overall structure and background in the source image while the second term facilitates alignment with the target domain by selectively editing the regions of interest. We visualize text-driven image-to-image translation results in Fig. 1, which demonstrates the outstanding performance of the proposed approach across various tasks.

The main contributions of our work are summarized below:

- We propose a novel approach to revise the standard noise prediction network by utilizing the prompt interpolation, which progressively updates the text embedding toward given target prompts during the reverse process of diffusion models.
- We formulate the proposed noise prediction network using two terms, which is coherent to the conceptual procedure of our task. One is the standard noise prediction term given the source image and the source prompt to reconstruct the overall structure and background, and the other is a new correction term

using the progressive prompt interpolation to selectively modify regions of interest.
- Experimental results demonstrate that our proposed method achieves remarkable translation results with time-efficient inference and improves the performance consistently when combined with existing methods.

The rest of our paper is organized as follows. Section 2 reviews the related work about text-driven image-to-image translation based on diffusion models. Section 3 describes the standard DDIM-based text-driven image-to-image translation algorithm, and Section 4 presents our approach. Our experimental results are provided in Section 5, and we finally conclude our paper in Section 6.

2   Related Work

This section discusses previous works about diffusion-based text-to-image generation and text-driven image-to-image editing approaches.

2.1   Text-to-Image Generation based on Diffusion Models

Diffusion-based text-to-image generation models [19, 21, 23] are typically trained on large-scale training datasets with image-caption pairs. Motivated by two-stage frameworks [3, 30], Stable Diffusion [21] projects input images onto a low-dimensional space using a pretrained autoencoder and a diffusion model learns to generate the low-dimensional features conditioned text embeddings given by a text encoder. DALLE-2 [19] first learns a prior model to estimate CLIP [17] image embeddings based on text captions and then employs a decoder to synthesize output images given the image features and their corresponding text captions. In contrast, Imagen [23] utilizes language models [18] to extract text features and learns text-to-image diffusion models to generate images conditioned on the text embeddings.

2.2   Text-Driven Image Editing based on Diffusion Models

The goal of text-driven image-to-image translation is to edit the specific regions in a source image to align a resulting target image with the target prompt while preserving the remaining parts. Existing text-driven image editing methods [1, 5, 9, 10, 16, 29] based on diffusion models are typically divided into two groups depending on whether they require an additional training or not. For example, DiffusionCLIP [10] fine-tunes a text-to-image diffusion model using the directional CLIP loss [4] for fidelity and the identity loss for preserving the background. Imagic [9] optimizes a pretrained diffusion model to reconstruct the source images conditioned on its predicted source text embedding while generating target images based on the interpolation between predicted source text embeddings and target text embeddings.

On the other hand, training-free image-to-image translation approaches [5, 16, 29] revise the reverse process of pretrained diffusion models. For instance,

Prompt-to-Prompt [5] and Plug-and-Play [29] inject the internal representations of source image—in the forms of cross-attention maps [5] or self-attention maps (and simple feature maps) [29]—into the target generation module. Pix2Pix-Zero [16] optimizes target latents by aligning the internal representations corresponding to the target and source latents and concurrently generates images with the optimized target latents using the original reverse process. Besides, diffusion-based image reconstruction techniques such as Null-text Inversion [15] and Negative-prompt Inversion [14] can be combined with existing image-to-image translation methods to improve performance, but they are not standalone translation methods.

The proposed approach revises the reverse process of diffusion models without any modification of the text-to-image diffusion backbones. Different from existing frameworks [5, 16, 29], we propose a simple but effective method to adjust the noise prediction network for text-driven image-to-image translation. Since our algorithm is orthogonal to existing methods, we empirically investigate the potential of our approach for performance improvement by combining it with the existing methods.

3   Text-Driven Image-to-Image Translation

This section describes the standard DDIM-based text-driven image editing approach, which consists of two deterministic processes: the inversion of a source image and the translation to the target domain.

3.1   Inference of Latent Variables for Source Images

Denoising diffusion probabilistic Models (DDPM) [7, 25] assume a Markovian stochastic process with Gaussian transition kernels, where $\mathbf{x}_0$ is a random variable for an image and $(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T)$ denotes a sequence of latent variables representing intermediate outputs in a diffusion process. Instead of using DDPM, existing text-driven image-to-image translation methods often rely on the deterministic DDIM inference [26] to reduce the number of inference steps without sacrificing the quality of generated images. Utilizing the denoising network denoted by $\epsilon_\theta(\cdot, \cdot, \cdot)$ which is parametrized with the U-Net architecture [22], the forward process of DDIM is formally given by

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

where $\mathbf{x}_t^{\mathrm{src}}$ is a source latent at a time step t, $\mathbf{y}^{\mathrm{src}}$ is the CLIP text embedding of the source prompt p^{src}, and α_t is a constant decreasing monotonically over time. From the above equation, $f_\theta(\cdot, \cdot, \cdot)$ is derived as

$$f_\theta(\mathbf{x}_t, t, \mathbf{y}) = \frac{\mathbf{x}_t - \sqrt{1 - \alpha_t} \epsilon_\theta(\mathbf{x}_t, t, \mathbf{y})}{\sqrt{\alpha_t}}. \tag{2}$$

Finally, $\mathbf{x}_T^{\mathrm{src}}$ is obtained from $\mathbf{x}_0^{\mathrm{src}}$ by recursively leveraging the deterministic DDIM forward process as described in Eq. (1), and is adopted for translating to the image in the target domain, $\mathbf{x}_0^{\mathrm{tgt}}$.
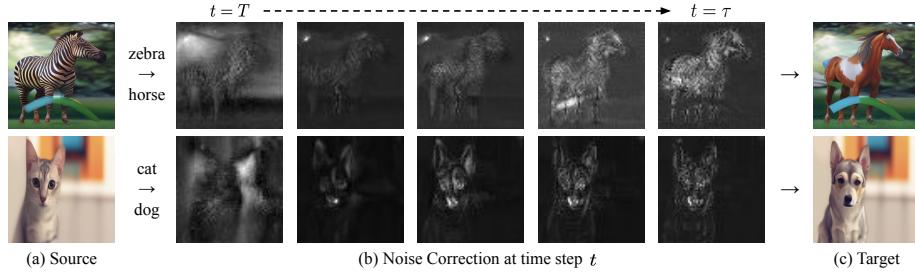
$t = T$ - → $t = \tau$

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

Fig. 2: Visualization of the progressively updated noise correction term $\Delta\epsilon_\theta(\mathbf{x}^{\mathrm{tgt}}, t, \mathbf{y}_t)$ over time for each pair of source and target images.

3.2   Reverse Process of Target Images

By simply setting $\mathbf{x}_T^{\mathrm{tgt}}$ equal to $\mathbf{x}_T^{\mathrm{src}}$, one can synthesize the target image using the following DDIM process [26]:

$$\mathbf{x}_{t-1}^{\mathrm{tgt}} = \sqrt{\alpha_{t-1}} f_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{tgt}}) + \sqrt{1 - \alpha_{t-1}} \epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{tgt}}), \qquad (3)$$

where $\mathbf{x}_t^{\mathrm{tgt}}$ is a target latent and $\mathbf{y}^{\mathrm{tgt}}$ is a CLIP feature of a target prompt p^{tgt}. However, the starting point of the reverse process, $\mathbf{x}_T^{\mathrm{tgt}}(= \mathbf{x}_T^{\mathrm{src}})$, is different from its true position $\mathbf{x}_T^{\mathrm{tgt}^*}$. Therefore, the naïve reverse process often fails to generate desired images in the target domain. The goal of our approach is to reroute the reverse process to compensate for its wrong initialization and successfully generate target images without additional training.

4   Our Approach

This section discusses how to improve the quality of translated images for text-driven image-to-image translation.

4.1   Overview

One of the reasons for poor image-to-image translation quality in naïve approaches is the abrupt transition of text embedding from $\mathbf{y}^{\mathrm{src}}$ to $\mathbf{y}^{\mathrm{tgt}}$ at the early stage in the reverse process. To address this issue, we formulate a noise prediction strategy for the text-driven image-to-image translation by progressively updating the text prompt embedding via time-dependent interpolations of the source and target prompt embeddings. We derive the revised version of the reverse process and introduce a correction term to update the convergence trajectory conditioned on the target prompt. Algorithm 1 presents the detailed procedure of the proposed method. We refer to the proposed algorithm as Prompt Interpolation-based Correction (PIC).

4.2   Noise Correction

To preserve the original structure or background in a source image, we compute a mixture representation of $\mathbf{y}^{\mathrm{src}}$ and $\mathbf{y}^{\mathrm{tgt}}$, which is given by

$$\mathbf{y}_t = h(\mathbf{y}^{\mathrm{src}}, \mathbf{y}^{\mathrm{tgt}}, t), \tag{4}$$

where $h(\cdot, \cdot, \cdot)$ is an interpolation function with a time-dependent mixing coefficient β_t, which will be discussed in Section 4.3. Then, we replace the original noise prediction network $\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{tgt}})$ with a new one, $\hat{\epsilon}_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{tgt}})$, which is given by

$$\hat{\epsilon}_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{tgt}}) := \epsilon_\theta(\mathbf{x}_t^{\mathrm{src}}, t, \mathbf{y}^{\mathrm{src}}) + \gamma \Delta\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}_t), \tag{5}$$

where γ is a hyperparameter and $\Delta\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}_t)$ is a correction term with the interpolated text prompt embedding $\mathbf{y}_t$. In this formulation, $\epsilon_\theta(\mathbf{x}_t^{\mathrm{src}}, t, \mathbf{y}^{\mathrm{src}})$ enables our approach to preserve the structure or background of a source image $\mathbf{x}_0^{\mathrm{src}}$ while the additional correction term facilitates the alignment of the generated image to the target domain.

Conceptually, it is desirable for the noise correction term, $\Delta\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}_t)$, to only affect the relevant regions to the target prompt while preserving the rest of the source image. The formal definition of the correction term $\Delta\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}_t)$ is as follows:

$$\Delta\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}_t) := \epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}_t) - \epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{src}}), \tag{6}$$

where $\mathbf{y}_t$ moves from $\mathbf{y}^{\mathrm{src}}$ to $\mathbf{y}^{\mathrm{tgt}}$ as t decreases. By plugging Eq. (6) into Eq. (5), we obtain the following noise prediction network:

$$\hat{\epsilon}_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{tgt}}) = \epsilon_\theta(\mathbf{x}_t^{\mathrm{src}}, t, \mathbf{y}^{\mathrm{src}}) + \gamma \left(\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}_t) - \epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{src}}) \right). \tag{7}$$

Our intuition behind the noise correction term is that the noise prediction given the target latent and the progressively interpolated text embedding effectively makes up for the gap between the unknown true initialization, $\mathbf{x}_T^{\mathrm{tgt}*}$ and its trivial surrogate, $\mathbf{x}_T^{\mathrm{tgt}}(= \mathbf{x}_T^{\mathrm{src}})$. We observe that this correction term is particularly helpful at the early stage of the reverse process and is not necessarily required for the rest of the iterations. Fig. 2 supports our intuition by visualizing the noise correction term during the reverse process; it gradually highlights the regions to be updated while the background area is set to negligible values.

4.3   Prompt Interpolation

We now describe the proposed prompt interpolation strategies with the source and target embeddings, designed for the slightly different two tasks of interest: word replacement and adding phrases.

Algorithm 1 Target image generation by PIC

1: **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
2: **for** $t \leftarrow 0, \cdots, T-1$ **do**
3:     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 $\epsilon_\theta(\mathbf{x}_t^{\mathrm{src}}, t, \mathbf{y}^{\mathrm{src}})$
4: **end for**
5: $\mathbf{x}_T^{\mathrm{tgt}} \leftarrow \mathbf{x}_T^{\mathrm{src}}$
6: **for** $t \leftarrow T, \cdots, T-\tau+1$ **do**
7:     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
8:     Compute $\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}_t)$ and $\epsilon_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{src}})$
9:     Obtain the revised model $\hat{\epsilon}_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{tgt}})$ using Eq. (7)
10:     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 $\hat{\epsilon}_\theta(\mathbf{x}_t^{\mathrm{tgt}}, t, \mathbf{y}^{\mathrm{tgt}})$
11: **end for**
12: **for** $t \leftarrow T-\tau, \cdots, 1$ **do**
13:     Obtain $\mathbf{x}_{t-1}^{\mathrm{tgt}}$ using Eq. (3)
14: **end for**
15: **Output:** target image $\mathbf{x}_0^{\mathrm{tgt}}$

Word replacement For word replacement, we consider the scenario that the tokens in the source prompt are replaced by other ones. For example, in the case of 'zebra → horse', if the source prompt is *'A zebra is lying on the grass.'*, the target prompt becomes *'A horse is lying on the grass'* by replacing 'zebra' with 'horse'. In this task, our simple linear prompt interpolation is given by

$$\mathbf{y}_t[\ell] = \beta_t \mathbf{y}^{\mathrm{tgt}}[\ell] + (1 - \beta_t)\mathbf{y}^{\mathrm{src}}[\ell], \tag{8}$$

where ℓ is a token index and the time-dependent coefficient β_t is set to

$$\beta_t := \beta + (1 - \beta) \times \frac{T - t}{T}, \tag{9}$$

where β is an initialization value between 0 and 1. Note that the interpolated embedding is progressively updated starting from the source prompt embedding to the target prompt embedding during the reverse process.

Adding phrases We consider another task that involves the addition of tokens. For instance, in the case of 'dog → dog with glasses', if the source prompt is *'A dog is lying on the grass'*, then the target prompt becomes *'A dog with glasses is lying on the grass'*. In this task, we have to match tokens between the source and target prompt embeddings for prompt interpolation, which is given by

$$\mathbf{y}_t[\ell] = \begin{cases} \mathbf{y}^{\mathrm{src}}[\ell], & \text{if } \ell < \ell_s \\ \mathbf{y}^{\mathrm{tgt}}[\ell], & \text{if } \ell_s \leq \ell \leq \ell_f \\ \beta_t \mathbf{y}^{\mathrm{tgt}}[\ell] + (1 - \beta_t)\mathbf{y}^{\mathrm{src}}[\ell - \ell_f + \ell_s], & \text{if } \ell > \ell_f \end{cases} \tag{10}$$

where $n \, (= \ell_f - \ell_s + 1)$ tokens are inserted at the ℓ_s^{th} position of the source prompt and β_t is defined in Eq. (9). Note that this strategy interpolates the embeddings of the source and target prompts from the next token positions of the added phrase[1]

4.4   Integration into Existing Methods

The proposed technique can be conveniently incorporated into state-of-the-art methods for diffusion-based image-to-image translation such as Prompt-to-Prompt [5], Plug-and-Play [29], and Pix2Pix-Zero [16]. The algorithm-specific noise prediction network, $\hat{\epsilon}_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}^{\text{tgt}})$, derived from Eq. (5) is expressed as

$$\hat{\epsilon}_\theta^{\text{alg}}(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}^{\text{tgt}}) := \epsilon_\theta(\mathbf{x}_t^{\text{src}}, t, \mathbf{y}^{\text{src}}) + \gamma \Delta\epsilon_\theta^{\text{alg}}(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}_t), \qquad (11)$$

where $\Delta\epsilon_\theta^{\text{alg}}(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}_t)$ is the noise correction term, specific to the individual translation algorithms [5, 16, 29]. The rest of this subsection discusses how to obtain the new noise correction term $\Delta\epsilon_\theta^{\text{alg}}(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}_t)$ for each algorithm.

Prompt-to-Prompt [5] The extension of the proposed prompt interpolation technique to Prompt-to-Prompt is simple. During the reverse process, Prompt-to-Prompt replaces the cross-attention and self-attention maps in $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}^{\text{tgt}})$ with the matching attention maps in $\epsilon_\theta(\mathbf{x}_t^{\text{src}}, t, \mathbf{y}^{\text{src}})$. Different from the vanilla Prompt-to-Prompt, our extension replaces the attention maps in $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}_t)$ to ones in $\epsilon_\theta(\mathbf{x}_t^{\text{src}}, t, \mathbf{y}^{\text{src}})$, instead of $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}^{\text{tgt}})$.

Plug-and-Play [29] Plug-and-Play performs the reverse process with the substitution of the self-attention maps and the intermediate feature maps in the denoising network $\epsilon_\theta(\mathbf{x}_t^{\text{src}}, t, \mathbf{y}^{\text{src}})$ for those obtained from $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}^{\text{tgt}})$. As in our extension to Prompt-to-Prompt, we use $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}_t)$ instead of $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}^{\text{tgt}})$ to compute the attention and feature maps for the replacements.

Pix2Pix-Zero [16] For the reverse process, Pix2Pix-Zero obtains the target latent $\hat{\mathbf{x}}_t^{\text{tgt}}$ by taking a gradient step from $\mathbf{x}_t^{\text{tgt}}$ using the cross-attention guidance loss, which aims to align the cross attention maps in the denoising network given the source and the target latents. The optimized target latent is given by

$$\hat{\mathbf{x}}_t^{\text{tgt}} = \mathbf{x}_t^{\text{tgt}} - \lambda_{\text{xa}} \nabla_{\mathbf{x}_t^{\text{tgt}}} \|\mathbf{M}_t^{\text{tgt}} - \mathbf{M}_t^{\text{src}}\|_F^2. \qquad (12)$$

where $\mathbf{M}_t^{\text{tgt}}$ and $\mathbf{M}_t^{\text{src}}$ denote the cross attention maps in $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}_t)$ and $\epsilon_\theta(\mathbf{x}_t^{\text{src}}, t, \mathbf{y}^{\text{src}})$. Respectively, λ_{xa} is a hyperparameter, and $\|\cdot\|_F$ indicates the Frobenius norm. Note that vanilla Pix2Pix-Zero obtains $\mathbf{M}_t^{\text{tgt}}$ from $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}^{\text{tgt}})$. Therefore, the noise correction term specific to Pix2Pix-Zero, is given by

$$\Delta\epsilon_\theta^{\text{P2P}}(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}_t) := \epsilon_\theta(\hat{\mathbf{x}}_t^{\text{tgt}}, t, \mathbf{y}_t) - \epsilon_\theta(\hat{\mathbf{x}}_t^{\text{tgt}}, t, \mathbf{y}^{\text{src}}), \qquad (13)$$

where $\mathbf{x}_t^{\text{tgt}}$ is replaced by $\hat{\mathbf{x}}_t^{\text{tgt}}$ from Eq. (6).

---

[1] Our prompt interpolation strategy for adding phrases can be generalized to the cases where phrases are removed.

Table 1: Quantitative comparisons of PIC with Prompt-to-Prompt [5], Plug-and-Play [29], and Pix2Pix-Zero [16] on images sampled from the LAION-5B dataset [24] using the pretrained Stable Diffusion [21] backbone. Black and red bold-faced numbers denote the best and second-best performances within each row for each metric.

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**

Table 2: Quantitative comparisons of the proposed method with Prompt-to-Prompt [5] on images sampled from the LAION-5B dataset [24] using the pretrained Stable Diffusion [21]. Out technique is integrated into Prompt-to-Prompt and the results of Prompt-to-Prompt are obtained from Tab. 1. Black bold-faced numbers represent better performance on each metric between two approaches.

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**

5 Experiments

We evaluate the performance of our approach, PIC, in comparison with the state-of-the-art training-free diffusion-based image-to-image translation methods [5, 16, 29]. We identify the 250 most relevant images for the desired source domain given a task, based on their CLIP similarities, and use them as inputs for image-to-image translation methods to be tested in the task. Note that the algorithm integrating PIC is denoted by by '[Algorithm Name] + PIC'.

5.1 Implementation Details

We implement the proposed method using the publicly available code of Pix2Pix-Zero (P2P)[2]. We integrate PIC into the existing techniques—Prompt-to-Prompt (PtP)[3], Plug-and-Play (PnP)[4] and Pix2Pix-Zero (P2P)—using their official codes. To accelerate the text-driven image-to-image translation process, the inference time steps for the forward and reverse processes are set to 50. For all experiments, Stable Diffusion v1.4 is employed as the backbone model. During the forward

---

[2] https://github.com/pix2pixzero/pix2pix-zero

[3] https://github.com/google/prompt-to-prompt

[4] https://github.com/MichalGeyer/plug-and-play

Table 3: Quantitative comparisons of the proposed method with Plug-and-Play [29] on images sampled from the LAION-5B dataset [24] using the pretrained Stable Diffusion [21]. Out technique is integrated into Plug-and-Play and the results of Plug-and-Play are obtained from Tab. 1.

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**

Table 4: Quantitative comparisons of the proposed method with Pix2Pix-Zero [16] on images sampled from the LAION-5B dataset [24] using the pretrained Stable Diffusion [21]. Out technique is integrated into Pix2Pix-Zero and the results of Pix2Pix-Zero are obtained from Tab. 1.

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**

process, we adopt Bootstrapping Language-Image Pretraining (BLIP) [12] to generate a source prompt for conditioning the denoising network. The target prompt is given by replacing the specific words in the source prompt with the alternatives defined by an assigned task as mentioned in Section 4.3. We use the same source and target prompts of all algorithms for the fair comparisons during both the forward and reverse processes. Additionally, we adopt classifier-free guidance [8] following [5, 16, 29].

In our implementation, τ and γ are set to 25 and 1.0, respectively, for all experiments. Also, we set β to 0.3 for word replacement tasks (*e.g.* 'dog → cat' and 'horse → zebra') while it is set to 0.8 for adding phrases tasks (*e.g.* 'tree → palm tree' and 'dog → dog with glasses').

5.2   Evaluation Metrics

For quantitative evaluation, we measure CLIP Similarity [6], Background Distance, and Structure Distance [28] following Pix2Pix-Zero [16]. The CLIP similarity (CS) quantifies how well the translated images are aligned with the target prompts using the cosine similarity. On the other hand, the background distance (BD) calculates the Learned Perceptual Image Patch Similarity (LPIPS) score [31] between the background regions of the source and translated images. To identify background regions, we employ the prediction of the pretrained ob-

Fig. 3: Qualitative comparisons between PIC and state-of-the-art methods [5,16,29] on images from LAION-5B [24] using the pretrained Stable Diffusion [21]. PIC generates target images with higher-fidelity than others in all tasks. Note that all algorithms fail to preserve pose and texture of the source image in the last task, but PIC still shows a favorable result.
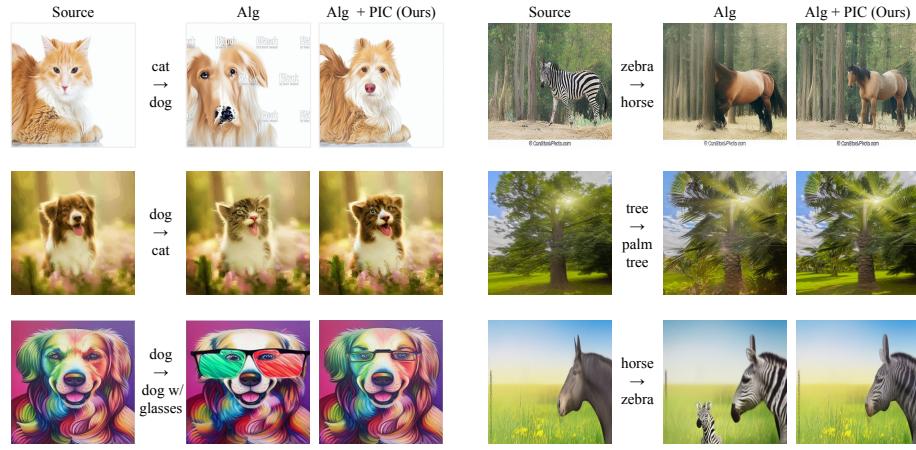
Fig. 4: Qualitative results of existing state-of-the-art methods and their combinations with PIC based on the pretrained Stable Diffusion [21]: (top) Prompt-to-Prompt [5], (middle) Plug-and-Play [29], and (bottom) Pix2Pix-Zero [16]. The examples are sampled from LAION-5B [24].

ject detector [20]. Also, the structure distance (SD) is employed to evaluate the structural difference between the source and translated images. It computes the Frobenius norm between the self-attention maps given by the DINO-ViT network output [2] using the source and translated images as inputs.

5.3   Quantitative Results

To compare the proposed method with state-of-the-art methods [5, 16, 29], we present quantitative results in Tab. 1. The table shows that our method consistently achieves the best performance in terms of BD and mostly outperforms the previous methods in terms of SD. As for CS, the proposed method shows the highest performance on the dog → cat task, while it ranks second in the remaining tasks. Note that, because the CLIP similarity only reflects the fidelity to the target prompt without considering the similarity to the source images, it is not sufficiently discriminative to evaluate image-to-image translation performance by itself. In addition, Tabs. 2 to 4 demonstrate that PIC is effective to improve the performance when incorporated into existing methods [5, 16, 29].

5.4   Qualitative Results

Fig. 3 illustrates qualitative results generated by the proposed approach and other state-of-the-art methods [5, 16, 29]. It presents that our method effectively preserves the background and structure of source images while selectively editing the region of interest. On the other hand, existing algorithms often fail to preserve the structure or background. We present a failure case of our algorithm

Table 5: Inference time comparisons between PIC and other state-of-the-art methods [5, 16, 29].

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

Table 6: Contribution of the noise correction and the prompt interpolation tested on LAION-5B dataset [24]. DDIM+PI synthesizes target images by replacing $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}^{\text{tgt}})$ with $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}_t)$ in the reverse DDIM process. The model with the noise correction, DDIM+NC, substitutes $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}^{\text{tgt}})$ for $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}_t)$ without the consideration of the prompt interpolation.

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**	0.312	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

in the last row of Fig. 3, where the result from PIC is still favorable compared to others. Fig. 4 demonstrates that PIC is effective to improve the previous methods when integrated into them.

5.5 Inference Time

To evaluate the inference time of each algorithm, we measure the wall-clock time using a single image on an NVIDIA A6000 GPU. As shown in Tab. 5, PIC is the most time-efficient even with its outstanding performance.

5.6 Ablation Study

Prompt Interpolation To analyze the impact of each component in our algorithm, we compare PIC with its three variations—DDIM, DDIM+PI, and DDIM+NC. DDIM denotes a naïve application of the original DDIM algorithm [26] to image-to-image translation. DDIM+PI replaces the denoising network $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}^{\text{tgt}})$ in Eq. (3) with $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}_t)$ using interpolated prompts $\mathbf{y}_t$ while DDIM+NC substitutes $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}^{\text{tgt}})$ for $\epsilon_\theta(\mathbf{x}_t^{\text{tgt}}, t, \mathbf{y}_t)$ in Eq. (7) to compute the noise correction term without the proposed prompt interpolation. As presented in Tab. 6, DDIM+PI improves performance by using prompt interpolation compared with the standard DDIM and DDIM+NC is particularly helpful in preserving the background or structure of the source images by integrating the noise correction term. Our algorithm, PIC, incorporating both the noise correction term and the prompt interpolation, achieves the best performance in the text-conditional image editing task. The qualitative results are presented in Fig. 6 of the appendix.
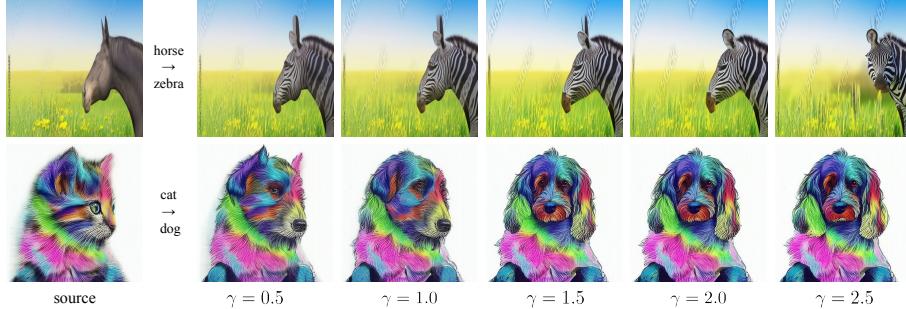
Fig. 5: Qualitative results of the proposed method by varying γ on data sampled from the LAION-5B dataset [24], relying on the pretrained Stable Diffusion [21].

Effect of Hyperparameter γ We study the effect of the hyperparameter γ introduced in Eq. (7) to discuss the trade-off between the fidelity to the target prompt and the structure preservation.

For the experiment related to γ, we explore five different values of $\gamma \in \{0.5, 1.0, 1.5, 2.0, 2.5\}$ for PIC. Fig. 5 illustrates that our results are fairly consistent to the value of γ. However, we observe that a low value of γ tends to preserve the structure or background with relatively low fidelity, while a high value of γ enhances fidelity at the expense of structure deformation. Note that we use $\gamma = 1.0$ throughout all experiments.

6   Conclusion

We presented a novel training-free approach for image-to-image translation based on text-to-image diffusion models. We revised the original noise prediction network by incorporating a noise correction term with progressive interpolation of text embeddings. Technically, the proposed noise prediction network for image-to-image translation consists of two parts: (a) the denoising network given the source latent and the source prompt and (b) a noise correction term defined as the difference between two noise predictions of the target latent conditioned on the progressively interpolated text embeddings and the source text embeddings. Extensive experiments demonstrate that the proposed algorithm achieves outstanding performance with reduced inference time and consistently improves existing techniques through the combination of those methods.

Acknowledgements

References

1. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to Follow Image Editing Instructions. In: CVPR (2023)
2. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging Properties in Self-Supervised Vision Transformers. In: ICCV (2021)
3. Esser, P., Rombach, R., Ommer, B.: Taming Transformers for High-Resolution Image Synthesis. In: CVPR (2021)
4. Gal, R., Patashnik, O., Maron, H., Bermano, A.H., Chechik, G., Cohen-Or, D.: StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators. TOG (2022)
5. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-or, D.: Prompt-to-Prompt Image Editing with Cross-Attention Control. In: ICLR (2023)
6. Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., Choi, Y.: CLIPScore: A Reference-free Evaluation Metric for Image Captioning. In: EMNLP (2021)
7. Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models. In: NeurIPS (2020)
8. Ho, J., Salimans, T.: Classifier-Free Diffusion Guidance. In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications (2021)
9. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based Real Image Editing with Diffusion Models. In: CVPR (2023)
10. Kim, G., Kwon, T., Ye, J.C.: DiffusionCLIP: Text-Guided Diffusion Models for Robust Image Manipulation. In: CVPR (2022)
11. Lee, H., Kang, M., Han, B.: Conditional Score Guidance for Text-Driven Image-to-Image Translation. In: NeurIPS (2023)
12. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML (2022)
13. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In: ICLR (2022)
14. Miyake, D., Iohara, A., Saito, Y., Tanaka, T.: Negative-prompt Inversion: Fast Image Inversion for Editing with Text-guided Diffusion Models. arXiv preprint arXiv:2305.16807 (2023)
15. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text Inversion for Editing Real Images Using Guided Diffusion Models. In: CVPR (2023)
16. Parmar, G., Kumar Singh, K., Zhang, R., Li, Y., Lu, J., Zhu, J.Y.: Zero-Shot Image-to-Image Translation. In: SIGGRAPH (2023)
17. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning Transferable Visual Models from Natural Language Supervision. In: ICML (2021)
18. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. JMLR (2020)
19. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
20. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., et al.: Grounded sam: Assembling open-world models for diverse visual tasks. arXiv preprint arXiv:2401.14159 (2024)

21. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-Resolution Image Synthesis with Latent Diffusion Models. In: CVPR (2022)
22. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: MICCAI (2015)
23. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In: NeurIPS (2022)
24. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. In: NeurIPS Datasets and Benchmarks Track (2022)
25. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In: ICML (2015)
26. Song, J., Meng, C., Ermon, S.: Denoising Diffusion Implicit Models. In: ICLR (2021)
27. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-Based Generative Modeling through Stochastic Differential Equations. In: ICLR (2021)
28. Tumanyan, N., Bar-Tal, O., Bagon, S., Dekel, T.: Splicing ViT Features for Semantic Appearance Transfer. In: CVPR (2022)
29. Tumanyan, N., Geyer, M., Bagon, S., Dekel, T.: Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation. In: CVPR (2023)
30. Van Den Oord, A., Vinyals, O., et al.: Neural Discrete Representation Learning. In: NIPS (2017)
31. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In: CVPR (2018)