A Latent Space of Stochastic Diffusion Models for Zero-Shot Image Editing and Guidance

Chen Henry Wu, Fernando De la Torre
Robotics Institute, Carnegie Mellon University
{chenwu2,ftorre}@cs.cmu.edu

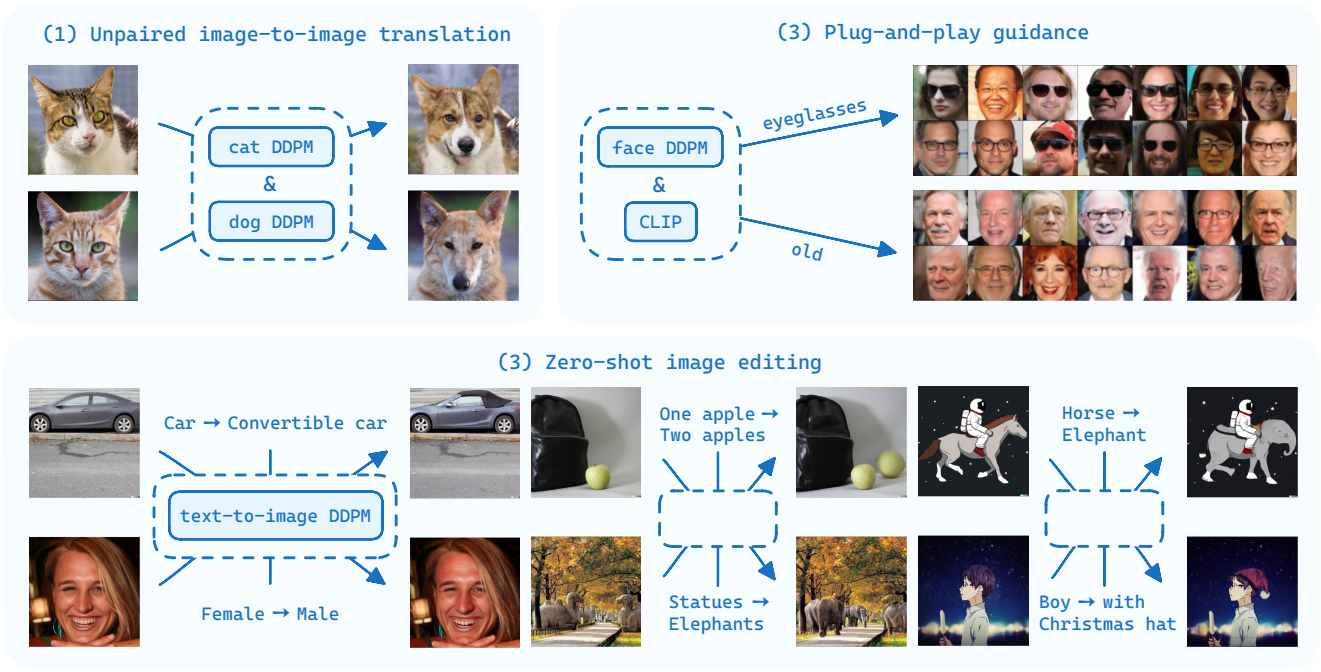


Figure 1: We define a new latent space for *stochastic* diffusion probabilistic models. This latent space allows us to perform: (1) Unpaired image-to-image translation with two diffusion models pre-trained independently (e.g., cat and dog). In this case, we can transfer the texture characteristics from an image of a cat to the model of a dog in an unsupervised fashion. (2) zero-shot image editing with a pre-trained text-to-image diffusion model. In this case, we can edit images with text prompts. (3) Plug-and-play guidance of a pre-trained diffusion model with off-the-shelf image understanding models such as CLIP. In this case, we are able to sub-sample a generative model of faces guided by attributes like "eyeglasses" or "old".

Abstract

Diffusion models generate images by iterative denoising. Recent work has shown that by making the denoising process deterministic*, one can encode real images into latent codes of the same size, which can be used for image editing. This paper explores the possibility of defining a latent space even when the denoising process remains* stochastic*. Recall that, in stochastic diffusion models, Gaussian noises are added in each denoising step, and we can concatenate all the noises to form a latent code. This results in a latent space of much higher dimensionality than the original image. We demonstrate that this latent space of stochastic diffusion models can be used in the same way as that of deterministic diffusion models in two applications. First, we propose CycleDiffusion, a method for zero-shot and unpaired image editing using stochastic diffusion models, which improves the performance over its deterministic counterpart. Second, we demonstrate unified, plug-and-play guidance in the latent spaces of deterministic and stochastic diffusion models.*[1]

[1]The code is available at humansensinglab/cycle-diffusion.

1. Introduction

Diffusion probabilistic models (DPMs) [30, 9] have achieved unprecedented results in generative modeling and are instrumental to text-to-image models such as DALL·E-2 [24]. In DPMs, images are generated by iterative denoising. In the original formulation, the denoising process is stochastic, where Gaussian noises are added in each denoising step. This stochastic formulation makes DPMs different from previous models such as VAEs [15], GANs [6], and normalizing flows [14], for which the generation can be inverted to obtain a latent code from a real image.

Different from the stochastic formulation, prior works [31, 29] showed that every stochastic DPM has an ODE-based, deterministic counterpart with the same output distribution. An important advantage of ODE-based, deterministic DPMs is that the denoising process can be inverted. That is, given an image, one can obtain a latent code (of the same size as the image) that can be denoised to reconstruct the image. This property makes deterministic DPMs a promising candidate for image editing [32], where one can encode an image into a latent code with one model (or condition) and decode it with another model (or condition) to obtain a new image.

In this paper, we show that (1) stochastic DPMs can also have a latent space, (2) real images can be encoded into this latent space, and (3) the latent space can be used in the same way as the latent space of deterministic DPMs.

To define the latent space, recall that in stochastic DPMs, Gaussian noises are added in each denoising step, and we concatenate all the noises to form a latent code. This results in a latent space of much higher dimensionality than the original image. For encoding, we propose DPM-Encoder, an algorithm for encoding real images into the latent space of stochastic DPMs. DPM-Encoder is based on the fact that the process of adding noises to the real image (i.e., the forward process) is pre-defined. Therefore, we can sample consecutive noisy images from the forward process and compute the noise used in denoising *by definition*.

We will show two cases in which this latent space of stochastic DPMs can be used in the same way as the latent space of deterministic DPMs. First, previous works [32, 7] have shown that deterministic DPMs can be used for image editing. Given two deterministic DPMs trained with the same noise schedule on two domains, one can encode a source image into a latent code with the source-domain model and decode it with the target-domain model to obtain a target image [32]. Given a deterministic text-to-image diffusion model, one can encode an image into a latent code conditioned on the source text and decode conditioned on the target text to obtain a new image [7]. Built upon these works, we propose CycleDiffusion, an extension of the same idea from deterministic DPMs to stochastic DPMs. Our experiments show that CycleDiffusion outperforms previous methods for both unpaired image editing (Section 4.1) and zero-shot image editing (Section 4.2).

Second, we show that both deterministic and stochastic DPMs can be guided in a plug-and-play manner using the energy-based model [18, 20, 35]. Such plug-and-play guidance was previously proposed to sample controlled distributions from pre-trained GANs. Notably, different from classifier guidance [5], the plug-and-play guidance does not require finetuning the guidance model on noisy images. Our experiments demonstrate that plug-and-play guidance can sample controlled and diverse images from both deterministic and stochastic DPMs (Section 4.3).

2. Related Work

Diffusion models for image generation Diffusion probabilistic models (DPMs) [30, 9] are a class of generative models that generate images by iterative denoising, which have been the basis of text-to-image synthesis such as DALL·E-2 [24] and Stable Diffusion [25]. Once the diffusion model is trained, different denoising processes can be used to generate images, which theoretically results in the same marginal output distribution. The denoising process can be categorized into two types: stochastic and deterministic. In stochastic DPMs [30, 9], Gaussian noises are added in each denoising step, while in deterministic DPMs [31, 29], the denoising process is a deterministic mapping from latent codes to images. One benefit of deterministic DPMs is that the denoising process can be inverted based on its ODE formulation [31, 29]. The inverted latent code can then be used for applications such as real image editing [32]. In this paper, we show that we can also define a latent space for stochastic DPMs, which can be used in the same way as the latent space of deterministic DPMs.

Image editing with diffusion models Recent works have shown that diffusion models are capable of image editing [17, 32, 3, 7]. Among them, [32, 7, 13] are based on the observation that a fixed random seed can be used to generate images with minimal changes. However, in order to edit *real* images, they need to adopt the ODE-based deterministic DPMs to invert images into the latent space, which is not suitable for stochastic DPMs. To the best of our knowledge, our work is the first to show that stochastic DPMs can also be used for image editing. Furthermore, [7] shows that fixing the cross-attention map in Transformer-based text-to-image diffusion models further preserves the structure of images; we show that this can be done for stochastic DPMs as well.

Guiding diffusion models After the diffusion model is pre-trained, it can be guided by additional inputs to generate images with specific properties, such as class labels [5], text [19], and corrupted images [12]. Guidance methods such as classifier guidance [5] and CLIP guidance [19, 16]

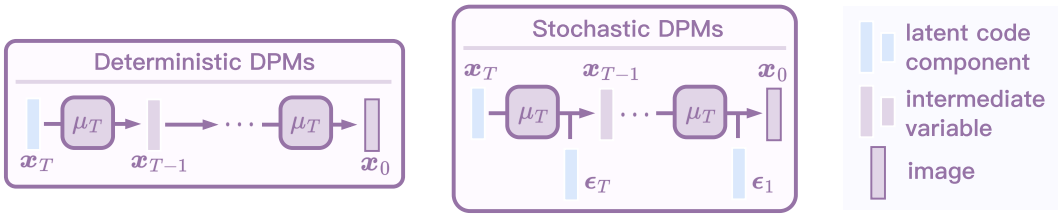


Figure 2: Both deterministic and stochastic DPMs can be formulated as deterministic maps from latent code $\boldsymbol{z}$ to image $\boldsymbol{x}$.

requires require the classifier and CLIP to be trained or finetuned on noisy images at all noise levels, meaning that the guidance is not plug-and-play. Classifier-free guidance (CFG) [10] does not require finetuning classifiers or CLIP, while it only applies to models that are already conditional. In this work, we show that the latent space formulation allows deterministic and stochastic DPMs to be guided in a plug-and-play manner, in the same way as previous works on GANs [18, 20, 35].

3. Method

3.1. Diffusion Models

Diffusion probablistic models (DPMs) [9, 30] generate images by iterative denoising. Two types of diffusion models have been proposed: stochastic DPMs and deterministic DPMs. Stochastic DPMs [9, 30, 29] generate images with a Markov chain structure. Given the mean estimator $\boldsymbol{\mu}_T$ (usually parameterized as a UNet), the image $\boldsymbol{x} := \boldsymbol{x}_0$ is generated through the following process:

$$\begin{aligned} \boldsymbol{x}_T &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \\ \boldsymbol{x}_{t-1} &= \boldsymbol{\mu}_T(\boldsymbol{x}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t, \\ \boldsymbol{\epsilon}_t &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad t = T, \dots, 1. \end{aligned} \tag{1}$$

Deterministic DPMs [29, 31, 11] generate images with the ODE formulation, which remove the randomness during the denoising process. Given the estimator $\boldsymbol{\mu}_T$ (usually parameterized as a UNet) for the denoising direction, deterministic DPMs generate $\boldsymbol{x} := \boldsymbol{x}_0$ via

$$\begin{aligned} \boldsymbol{x}_T &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \\ \boldsymbol{x}_{t-1} &= \boldsymbol{\mu}_T(\boldsymbol{x}_t, t), \quad t = T, \dots, 1. \end{aligned} \tag{2}$$

3.2. Gaussian Latent Space for Diffusion Models

Given an $\boldsymbol{x}_T$, an image $\boldsymbol{x}$ is generated from a deterministic DPM without any randomness, meaning that deterministic DPMs can be seen as a deterministic mapping $G : \mathbb{R}^d \to \mathcal{X}$ from latent codes $\boldsymbol{z}$ to images $\boldsymbol{x}$. Specifically, based on Eq. (2), it is trivial to define the latent code $\boldsymbol{z}$ and the mapping G as

$$\begin{aligned} \boldsymbol{z} := \boldsymbol{x}_T &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \\ \boldsymbol{x}_{t-1} &= \boldsymbol{\mu}_T(\boldsymbol{x}_t, t), \quad t = T, \dots, 1. \end{aligned} \tag{3}$$

In this paper, we show that stochastic DPMs can also be formulated as deterministic maps from latent codes $\boldsymbol{z}$ to images $\boldsymbol{x}$. Specifically, based on Eq. (1), we define the latent code $\boldsymbol{z}$ and the mapping G as

$$\begin{aligned} \boldsymbol{z} &:= \big(\boldsymbol{x}_T \oplus \boldsymbol{\epsilon}_T \oplus \cdots \oplus \boldsymbol{\epsilon}_1\big) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \\ \boldsymbol{x}_{t-1} &= \boldsymbol{\mu}_T(\boldsymbol{x}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t, \quad t = T, \dots, 1. \end{aligned} \tag{4}$$

An overview of the two types of diffusion models is shown in Figure 2. A benefit of the latent space of deterministic DPMs is that it is easy to invert the mapping G to obtain the latent code $\boldsymbol{z}$ from the image $\boldsymbol{x}$ using the ODE formulation. This enables image editing applications using deterministic DPMs [32, 7]. In the following, we show that it is also possible to sample latent codes $\boldsymbol{z}$ from the image $\boldsymbol{x}$ using stochastic DPMs.

3.3. DPM-Encoder

To sample latent codes $\boldsymbol{z}$ from the image $\boldsymbol{x}$ using stochastic DPMs, we propose DPM-Encoder. Given a pretrained stochastic DPM, DPM-Encoder does not require any additional training; instead, it is a way to sample latent codes by definition. Recall that, for each image $\boldsymbol{x} := \boldsymbol{x}_0$, stochastic DPMs define a posterior distribution $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ [9, 29]. Based on $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ and Eq. (4), we can directly derive $\boldsymbol{z} \sim \mathrm{DPMEnc}(\boldsymbol{z}|\boldsymbol{x}, G)$ as (see details in Appendix A)

$$\begin{aligned} \boldsymbol{x}_1, \dots, \boldsymbol{x}_{T-1}, \boldsymbol{x}_T &\sim q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0), \\ \boldsymbol{\epsilon}_t &= \big(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_T(\boldsymbol{x}_t, t)\big)/\boldsymbol{\sigma}_t, \quad t = T, \dots, 1, \\ \boldsymbol{z} &:= \big(\boldsymbol{x}_T \oplus \boldsymbol{\epsilon}_T \oplus \cdots \oplus \boldsymbol{\epsilon}_2 \oplus \boldsymbol{\epsilon}_1\big). \end{aligned} \tag{5}$$

A property of DPM-Encoder is perfect reconstruction, meaning that we have $\boldsymbol{x} = G(\boldsymbol{z})$ for every $\boldsymbol{z} \sim \mathrm{Enc}(\boldsymbol{z}|\boldsymbol{x}, G)$. A proof by induction is provided in Appendix A.

3.4. Application 1: Edit Images with CycleDiffusion

Given two stochastic DPMs G_1 and G_2 that model two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$, several researchers and practitioners have found that sampling with the same "random seed" leads to similar images [19]. Based on this finding, previous works [32, 7] have proposed to use deterministic DPMs for image editing. Specifically, one can use the ODE formulation to invert the mapping G to obtain the latent code $\boldsymbol{z}$

Algorithm 1: CycleDiffusion for zero-shot image editing, using a text-guided diffusion model

Input: source image $\boldsymbol{x} := \boldsymbol{x}_0$; source text $\boldsymbol{t}$; target text $\hat{\boldsymbol{t}}$; encoding step $T_{\text{es}} \leq T$

1. Sample noisy image $\hat{\boldsymbol{x}}_{T_{\text{es}}} = \boldsymbol{x}_{T_{\text{es}}} \sim q(\boldsymbol{x}_{T_{\text{es}}}|\boldsymbol{x}_0)$

for $t = T_{es}, \dots, 1$ **do**

2. $\boldsymbol{x}_{t-1} \sim q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$
3. $\boldsymbol{\epsilon}_t = \big(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_T(\boldsymbol{x}_t, t|\boldsymbol{t})\big)/\boldsymbol{\sigma}_t$
4. $\hat{\boldsymbol{x}}_{t-1} = \boldsymbol{\mu}_T(\hat{\boldsymbol{x}}_t, t|\hat{\boldsymbol{t}}) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t$

Output: $\hat{\boldsymbol{x}} := \hat{\boldsymbol{x}}_0$

from the image $\boldsymbol{x}$, and then use $\boldsymbol{z}$ to generate a new image $\hat{\boldsymbol{x}} = G(\boldsymbol{z})$. The encoding and decoding processes are using different model checkpoints or either conditioned on different text prompts.

In this work, we show that our definition of the latent space and DPM-Encoder allows stochastic DPMs to be used for image editing. Specifically, we propose a simple unpaired image-to-image translation method named CycleDiffusion. Given a source image $\boldsymbol{x} \in \mathcal{D}_1$, we use DPM-Encoder to encode it as $\boldsymbol{z}$ and then decode it as $\hat{\boldsymbol{x}} = G_2(\boldsymbol{z})$:

$$\boldsymbol{z} \sim \text{DPMEnc}(\boldsymbol{z}|\boldsymbol{x}, G_1), \quad \hat{\boldsymbol{x}} = G_2(\boldsymbol{z}). \tag{6}$$

We can also apply CycleDiffusion to text-to-image diffusion models by defining $\mathcal{D}_1$ and $\mathcal{D}_2$ as image distributions conditioned on two texts. Let $G_{\boldsymbol{t}}$ be a text-to-image diffusion model conditioned on text $\boldsymbol{t}$. Given a source image $\boldsymbol{x}$, the user writes two texts: a source text $\boldsymbol{t}$ describing the source image $\boldsymbol{x}$ and a target text $\hat{\boldsymbol{t}}$ describing the target image $\hat{\boldsymbol{x}}$ to be generated. We can then perform zero-shot image-to-image editing via (zero-shot means that the model has never been trained on image editing)

$$\boldsymbol{z} \sim \text{DPMEnc}(\boldsymbol{z}|\boldsymbol{x}, G_{\boldsymbol{t}}), \quad \hat{\boldsymbol{x}} = G_{\hat{\boldsymbol{t}}}(\boldsymbol{z}). \tag{7}$$

Inspired by the realism-faithfulness tradeoff in SDEdit [17], we can truncate $\boldsymbol{z}$ towards a specified encoding step $T_{\text{es}} \leq T$. The algorithm of CycleDiffusion is shown in Algorithm 1.

An analysis for image similarity with fixed $\boldsymbol{z}$. We analyze the image similarity using text-to-image diffusion models. Suppose the text-to-image model has two properties:

1. Conditioned on the same text, similar noisy images lead to similar enough mean predictions. Formally, $\boldsymbol{\mu}_T(\boldsymbol{x}_t, t|\boldsymbol{t})$ is K_t-Lipschitz, i.e., $\|\boldsymbol{\mu}_T(\boldsymbol{x}_t, t|\boldsymbol{t}) - \boldsymbol{\mu}_T(\hat{\boldsymbol{x}}_t, t|\boldsymbol{t})\| \leq K_t\|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\|$.
2. Given the same image, the two texts lead to similar predictions. Formally, $\|\boldsymbol{\mu}_T(\hat{\boldsymbol{x}}_t, t|\boldsymbol{t}) - \boldsymbol{\mu}_T(\hat{\boldsymbol{x}}_t, t|\hat{\boldsymbol{t}})\| \leq S_t$. Intuitively, a smaller difference between $\boldsymbol{t}$ and $\hat{\boldsymbol{t}}$ gives us a smaller S_t.

Let B_t be the upper bound of $\|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\|_2$ at time step t when the same latent code $\boldsymbol{z}$ is used for sampling (i.e., $\boldsymbol{x}_0 = G_{\boldsymbol{t}}(\boldsymbol{z})$ and $\hat{\boldsymbol{x}}_0 = G_{\hat{\boldsymbol{t}}}(\boldsymbol{z})$). We have $B_T = 0$ because $\|\boldsymbol{x}_T - \hat{\boldsymbol{x}}_T\|_2 = 0$, and B_0 is the upper bound for the generated images $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2$. The upper bound B_t can be propagated through time, from T to 0. Specifically, by combining the above two properties, we have

$$B_{t-1} \leq (K_t + 1)B_t + S_t. \tag{8}$$

3.5. Application 2: Plug-and-Play Guidance

Prior works showed that guidance for generative models can be achieved in the latent space [18, 20, 35]. Specifically, given a condition $\mathcal{C}$, one can define the guided image distribution as an energy-based model (EBM): $p(\boldsymbol{x}|\mathcal{C}) \propto p_{\boldsymbol{x}}(\boldsymbol{x})e^{-\lambda_{\mathcal{C}}E(\boldsymbol{x}|\mathcal{C})}$. Sampling for $\boldsymbol{x} \sim p(\boldsymbol{x}|\mathcal{C})$ is equivalent to $\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z}|\mathcal{C}), \boldsymbol{x} = G(\boldsymbol{z})$, where $p(\boldsymbol{z}|\mathcal{C}) \propto p_{\boldsymbol{z}}(\boldsymbol{z})e^{-\lambda_{\mathcal{C}}E(G(\boldsymbol{z})|\mathcal{C})}$. Examples of the energy function $E(\boldsymbol{x}|\mathcal{C})$ are provided in Section 4.3. To sample $\boldsymbol{z} \sim p(\boldsymbol{z}|\mathcal{C})$, one can use any model-agnostic samplers. For example, Langevin dynamics [34] starts from $\boldsymbol{z}^{\langle 0\rangle} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and samples $\boldsymbol{z} := \boldsymbol{z}^{\langle n\rangle}$ iteratively through

$$\boldsymbol{z}^{\langle k+1\rangle} = \boldsymbol{z}^{\langle k\rangle} + \frac{\sigma}{2}\nabla_{\boldsymbol{z}}\Big(\log p_{\boldsymbol{z}}(\boldsymbol{z}^{\langle k\rangle}) - E\big(G(\boldsymbol{z}^{\langle k\rangle})|\mathcal{C}\big)\Big) + \sqrt{\sigma}\boldsymbol{\omega}^{\langle k\rangle}, \tag{9}$$
$$\boldsymbol{\omega}^{\langle k\rangle} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}).$$

4. Experiments

This section provides experimental validation of the proposed work. Section 4.1 shows how CycleDiffusion achieves competitive results on unpaired image-to-image translation benchmarks. Section 4.2 provides a protocol for zero-shot image editing; CycleDiffusion outperforms several baselines. Section 4.3 presents results for controlling deterministic and stochastic DPMs in a plug-and-play manner.

4.1. Unpaired Image-to-Image Translation

Given two unaligned image domains, unpaired image-to-image translation maps images from one domain to the other. We follow setups from previous works whenever possible, as detailed below. Following [21, 37], we conducted experiments on the test set of AFHQ [4] with resolution 256×256 for Cat $\rightarrow$ Dog translation and Wild $\rightarrow$ Dog translation. For each source image, each method should generate a target image with minimal changes. After CycleDiffusion generates the translated image, we used T_{sdedit} steps of SDEdit to clean the artifacts. For $T = 1000$, we set $T_{\text{sdedit}} = 100$ for Cat $\rightarrow$ Dog and $T_{\text{sdedit}} = 125$ for Wild $\rightarrow$ Dog.

Metrics: To evaluate the realism of the generated image, we reported Frechet Inception Distance (FID) [8] and Kernel Inception Distance (KID) [2] between the generated and

	Cat → Dog				Wild → Dog			
	FID↓	KID×10^3↓	PSNR↑	SSIM↑	FID↓	KID×10^3↓	PSNR↑	SSIM↑
CUT (GAN SOTA [21])	76.21	–	17.48	0.601	92.94	–	17.20	0.592
ILVR [3]	74.37	–	17.77	0.363	75.33	–	16.85	0.287
SDEdit [17]	74.17	–	19.19	0.423	68.51	–	17.98	0.343
EGSDE [37]	65.82	–	**19.31**	0.415	59.75	–	**18.14**	0.343
CycleDiffusion w/ DDIM ($\eta = 0.1$)	**58.87**	20.3	18.50	**0.557**	**56.45**	19.5	17.82	**0.479**

Table 1: Quantitative comparison for unpaired image-to-image translation methods. Methods in the second block use the same pre-trained diffusion model in the target domain. Results of CUT, ILVR, SDEdit, and EGSDE are from [37]. Best results using diffusion models are in **bold**. CycleDiffusion has the best FID among all methods and the best SSIM among methods with diffusion models. Note that it has been shown that SSIM is much better correlated with human visual perception than squared distance-based metrics such as L_2 and PSNR [33].

Figure 3: Unpaired image-to-image translation (Cat → Dog, Wild → Dog) with CycleDiffusion.

target images. To evaluate faithfulness, we reported Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [33] between each generated image and its source image.

Baselines: We compared CycleDiffusion with previous state-of-the-art unpaired image-to-image translation methods: CUT [21], ILVR [3], SDEdit [17], and EGSDE [37]. CUT is based on GAN, and the others use diffusion models.

Pre-trained diffusion models: The baselines ILVR, SDEdit, and EGSDE only need the diffusion model trained on the target domain, and we followed them to use the pre-trained model from [3] for Dog. CycleDiffusion needs diffusion models on both domains, so we trained them on Cat and Wild.

Seen in Table 1 are the results. CycleDiffusion has the best realism (i.e., FID and KID). There is a mismatch between the faithfulness metrics (i.e., PSNR and SSIM), and note that SSIM is much better correlated with human perception than PSNR [33]. Among all diffusion model-based methods, CycleDiffusion achieves the highest SSIM. Figure 3 displays some image samples from CycleDiffusion, showing that our method can change the domain while preserving local details such as the background, lighting, pose, and overall color of the animal.

4.2. Zero-Shot Image Editing

This section provides the experiments for zero-shot image editing. We curated a set of 150 tuples $(\boldsymbol{x}, \boldsymbol{t}, \hat{\boldsymbol{t}})$ for this task, where $\boldsymbol{x}$ is the source image, $\boldsymbol{t}$ is the source text (e.g., "an aerial view of autumn scene." in Figure 4 second row on the right), and $\hat{\boldsymbol{t}}$ is the target text (e.g., "an aerial view of winter scene."). The generated image is denoted as $\hat{\boldsymbol{x}}$. At the end of this section, we also show that CycleDiffusion can be combined with the Cross Attention Control [7] to preserve the image structure.

Metrics: To evaluate the faithfulness of the generated image to the source image, we reported PSNR and SSIM. These metrics show how close the generated image is to the source image. To evaluate the authenticity of the generated image to the target text, we reported the CLIP score $\mathcal{S}_{\text{CLIP}} = \cos\left\langle \text{CLIP}_{\text{img}}(\hat{\boldsymbol{x}}), \text{CLIP}_{\text{text}}(\hat{\boldsymbol{t}}) \right\rangle$, where the CLIP embeddings are normalized. We note that there is a trade-off between PSNR/SSIM and $\mathcal{S}_{\text{CLIP}}$: by copying the source image without considering the text at all, one can get high PSNR/SSIM but low $\mathcal{S}_{\text{CLIP}}$; by ignoring the source image and directly generating images conditioned on the target text, one can get high $\mathcal{S}_{\text{CLIP}}$ but low PSNR/SSIM. To address this trade-off, we reported the directional CLIP score [22] (the

	Method	$\mathcal{S}_{\text{CLIP}}\uparrow$	$\mathcal{S}_{\text{D-CLIP}}\uparrow$	PSNR↑	SSIM↑
LDM-400M	SDEdit [17]	0.332	0.264	13.68	0.390
	DDIB [32]	0.324	0.195	15.82	0.544
	CycleDiffusion w/ DDIM ($\eta = 0.1$; ours)	**0.333**	**0.275**	**18.72**	**0.625**
Stable Diffusion v1-4	SDEdit [17]	**0.344**	0.258	15.93	0.512
	DDIB [32]	0.331	0.209	18.10	0.653
	CycleDiffusion w/ DDIM ($\eta = 0.1$; ours)	0.334	**0.272**	**21.92**	**0.731**

Table 2: Zero-shot image editing. **Protocol:** We did not use fixed hyperparameters, and neither did we plot the trade-off curve. The reason is that every input can have its best hyperparameters and even random seed. Instead, **for each input**, we ran 15 random trials for each hyperparameter and report the one with the highest $\mathcal{S}_{\text{D-CLIP}}$. For a fair comparison, different methods share the same set of combinations of hyperparameters if possible, detailed in Appendix B.

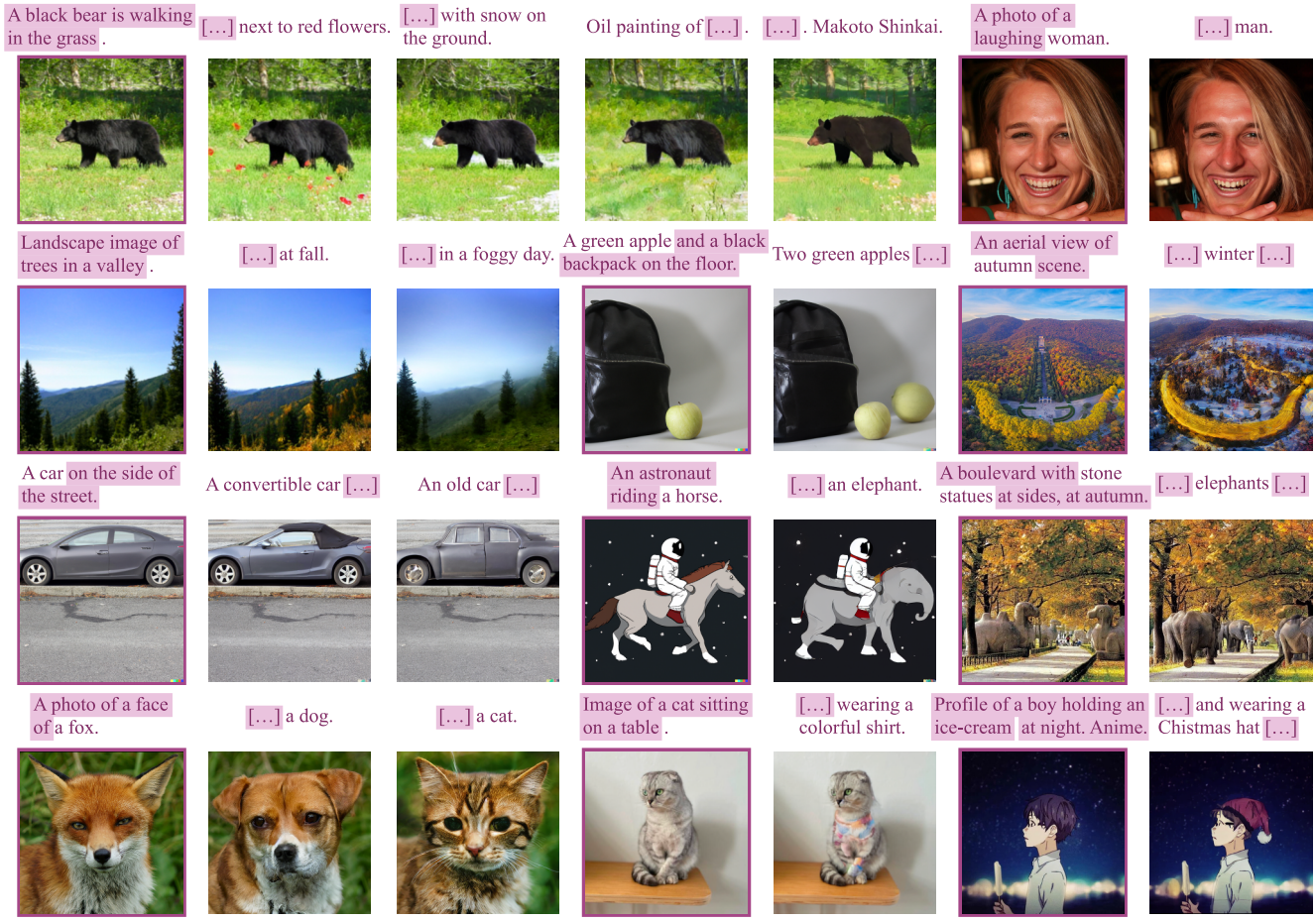


Figure 4: Examples of CycleDiffusion for zero-shot image editing. Source images $\boldsymbol{x}$ are displayed with a purple margin; the others are images $\hat{\boldsymbol{x}}$ generated by CycleDiffusion. Within each pair of source and target texts, overlapping text spans are marked in purple in the source text and abbreviated as […] in the target text.

CLIP embeddings are normalized):

$$\mathcal{S}_{\text{D-CLIP}} = \cos\Big\langle \text{CLIP}_{\text{img}}(\hat{\boldsymbol{x}}) - \text{CLIP}_{\text{img}}(\boldsymbol{x}), \\ \text{CLIP}_{\text{text}}(\hat{\boldsymbol{t}}) - \text{CLIP}_{\text{text}}(\boldsymbol{t}) \Big\rangle. \tag{10}$$

Baselines: We compared CycleDiffusion with two baselines: SDEdit [17] and DDIB [32]. SDEdit adds noises to the source image and denoise based on the target text; DDIB uses an ODE-based deterministic DPM to encode the source image to its latent code and then decodes the latent code

Figure 5: Visual comparison to the baselinses, DDIB and SDEdit. Notations follow Figure 4.

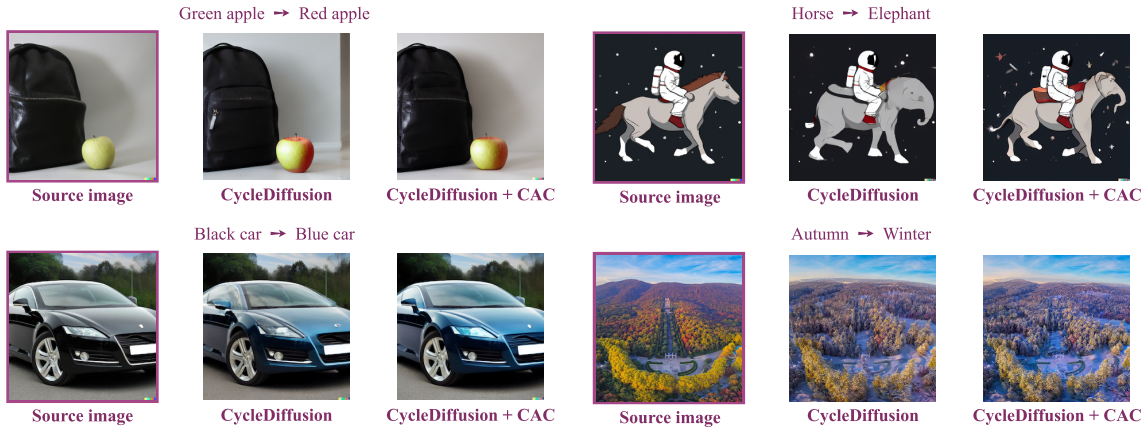
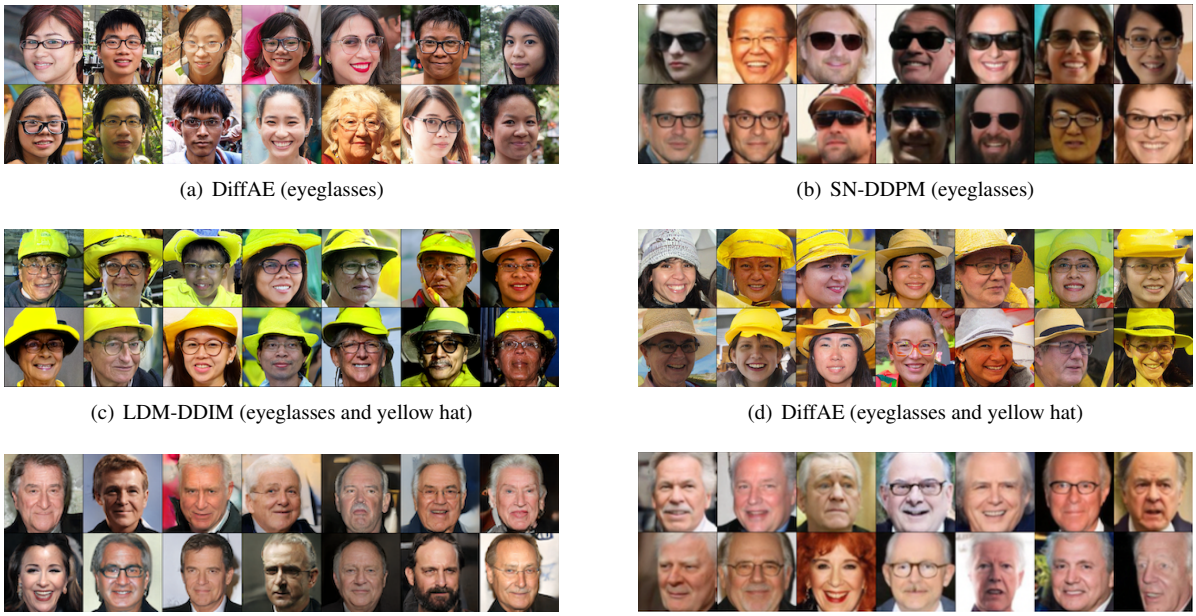


Figure 6: Cross Attention Control (CAC; [7]) helps CycleDiffusion when the intended *structural* change is small. For instance, when the intended change is color but not shape (left), CAC helps CycleDiffusion preserve the background; when the intended change is horse → elephant, CAC makes the generated elephant look more like a horse in shape.

conditioned on the target text. We used the same hyperparameters for the baselines and CycleDiffusion whenever possible (e.g., the number of diffusion steps, the strength of classifier-free guidance; see Appendix B).

Pre-trained text-to-image diffusion models: We used the following text-to-image diffusion models in our experiments: (1) `LDM-400M`, a 1.45B-parameter model trained on LAION-400M [28], (2) `Stable Diffusion v1-4`, a 0.98B-parameter model trained on LAION-5B [27].

Results: Table 2 shows the results for zero-shot image-to-image translation. CycleDiffusion excels at being faithful to the source image (i.e., PSNR and SSIM); by contrast, SDEdit and DDIB have comparable authenticity to the target text (i.e., $\mathcal{S}_{\text{CLIP}}$), but their outputs are much less faithful. Figure 4 provides samples from CycleDiffusion, demonstrating that CycleDiffusion achieves meaningful edits that span (1) replacing objects, (2) adding objects, (3) changing styles, and (4) modifying attributes. Figure 5 provides a qualitative comparison for zero-shot image-to-image translation. Compared with DDIB and SDEdit, CycleDiffusion greatly improves the faithfulness to the source image.



(a) DiffAE (eyeglasses)

(b) SN-DDPM (eyeglasses)

(c) LDM-DDIM (eyeglasses and yellow hat)

(d) DiffAE (eyeglasses and yellow hat)

(e) DDGAN (old person)

(f) SN-DDPM (old person)

Figure 7: Sampling text-conditioned sub-populations from diffusion models.

Compatibility with Cross Attention Control: Besides fixing the random seed, [7] shows that fixing the cross attention map (i.e., Cross Attention Control, or CAC) further improves the similarity between synthesized images. CAC is applicable to CycleDiffusion: in Algorithm 1, we can apply the attention map of $\boldsymbol{\mu}_T(\boldsymbol{x}_t, t|\boldsymbol{t})$ to $\boldsymbol{\mu}_T(\hat{\boldsymbol{x}}_t, t|\hat{\boldsymbol{t}})$. However, we cannot apply it to all samples because CAC puts requirements on the difference between $\boldsymbol{t}$ and $\hat{\boldsymbol{t}}$. Figure 6 shows that CAC helps CycleDiffusion when the intended *structural* change is small. For instance, when the intended change is color but not shape (left), CAC helps CycleDiffusion preserve the background; when the intended change is horse $\rightarrow$ elephant, CAC makes the generated elephant to look more like a horse in shape.

4.3. Plug-and-Play Guidance

Previous methods for guiding diffusion models, such as classifier guidance, require training the guidance model on noisy images [5, 16]. In this experiment, we explore how to sample text-conditioned images from diffusion models without finetuning the CLIP model on noisy images.

We instantiate the energy in Section 3.5 as $E_{\text{CLIP}}(\boldsymbol{x}|\boldsymbol{t}) = \frac{1}{L}\sum_{l=1}^{L}\Big(1 - \cos\big\langle \text{CLIP}_{\text{img}}\big(\text{DiffAug}_l(\boldsymbol{x})\big), \text{CLIP}_{\text{text}}(\boldsymbol{t})\big\rangle\Big)$, where DiffAug_l stands for differentiable augmentation [38] that mitigates the adversarial effect, and we sample from the energy-based distribution using Langevin dynamics in Eq. (9) with $n = 200$, $\sigma = 0.05$.

In Figure 7, we show several uncurated samples from different models and different text prompts (shown in parentheses). Among the methods, SN-DDPM [1] is a stochastic DPM, and LDM-DDIM [25] is a deterministic DPM. Besides these two, we also show other two variants of diffusion models, Denoising Diffusion GAN (DDGAN) [36] and Diffusion Autoencoder (DiffAE) [23]. These two models are not typical diffusion models, while they can also be formulated in the similar ways as in Figure 2. Results show that plug-and-play guidance is an effective way to guide deterministic and stochastic DPMs and their variants.

5. Conclusions

This paper proposes a latent space for stochastic diffusion models. Our idea is inspired by the latent space of deterministic diffusion models that allows interesting applications such as image editing. We show that our latent space for stochastic diffusion models can be used in the same way as that of deterministic diffusion models in several applications: (1) unpaired image-to-image translation with two diffusion models pre-trained independently (e.g., cat and dog); (2) zero-shot image editing with a pre-trained text-to-image diffusion model; (3) plug-and-play guidance of a pre-trained (stochastic or deterministic) diffusion model with off-the-shelf image understanding models such as CLIP. We hope that our work will inspire more research on understanding the generative process of stochastic diffusion models and their latent spaces.

References

[1] Fan Bao, Chongxuan Li, Jiacheng Sun, Jun Zhu, and Bo Zhang. Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. *ICML*, 2022.

[2] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. *ICLR*, 2018.

[3] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. ILVR: Conditioning method for denoising diffusion probabilistic models. *ICCV*, 2021.

[4] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. *CVPR*, 2020.

[5] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. *NeurIPS*, 2021.

[6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. *NIPS*, 2014.

[7] Amir Hertz, Ron Mokady, Jay M. Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *ArXiv*, 2022.

[8] 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. *NIPS*, 2017.

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

[10] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *NeurIPS Workshop*, 2021.

[11] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *NeurIPS*, 2022.

[12] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *NeurIPS*, 2022.

[13] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Hui-Tang Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *ArXiv*, 2022.

[14] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *NeurIPS*, 2018.

[15] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. *ICLR*, 2014.

[16] Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. More control for free! Image synthesis with semantic diffusion guidance. *ArXiv*, 2021.

[17] 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. *ICLR*, 2022.

[18] Anh M Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. *CVPR*, 2017.

[19] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. *ICML*, 2022.

[20] Weili Nie, Arash Vahdat, and Anima Anandkumar. Controllable and compositional generation with latent-space energy-based models. *NeurIPS*, 2021.

[21] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. *ECCV*, 2020.

[22] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and D. Lischinski. StyleCLIP: Text-driven manipulation of StyleGAN imagery. *ICCV*, 2021.

[23] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. *CVPR*, 2022.

[24] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *ArXiv*, 2022.

[25] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *CVPR*, 2022.

[26] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. *ArXiv*, 2022.

[27] Christoph Schuhmann, Romain Beaumont, Cade W Gordon, Ross Wightman, mehdi cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Richard Vencu, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. *NeurIPS Datasets and Benchmarks*, 2022.

[28] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: Open dataset of CLIP-filtered 400 million image-text pairs. *ArXiv*, 2021.

[29] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021.

[30] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 2019.

[31] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ICLR*, 2021.

[32] Xu Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. *ArXiv*, 2022.

[33] Zhou Wang, Alan Conrad Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004.

[34] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient Langevin dynamics. *ICML*, 2011.

[35] Chen Henry Wu, Saman Motamed, Shaunak Srivastava, and Fernando De la Torre. Generative visual prompt: Unifying distributional control of pre-trained generative models. *NeurIPS*, 2022.

[36] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. *ICLR*, 2022.

[37] Min Zhao, Fan Bao, Chongxuan Li, and Jun Zhu. EGSDE: Unpaired image-to-image translation via energy-guided stochastic differential equations. *NeurIPS*, 2022.

[38] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient GAN training. *NeurIPS*, 2020.

Algorithm 2: DPM-Encoder

Input: an image $\boldsymbol{x} := \boldsymbol{x}_0$, a pre-trained stochastic DPM with $\boldsymbol{\mu}_T(\boldsymbol{x}_t, t)$, $\boldsymbol{\sigma}_t$, and $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$

1. Sample $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{T-1}, \boldsymbol{x}_T \sim q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$
2. $\boldsymbol{z} = \boldsymbol{x}_T$

for $t = T, \dots, 1$ **do**

3. $\boldsymbol{\epsilon}_t = \big(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_T(\boldsymbol{x}_t, t)\big)/\boldsymbol{\sigma}_t$
4. $\boldsymbol{z} = \boldsymbol{z} \oplus \boldsymbol{\epsilon}_t$

5. **Output:** $\boldsymbol{z}$

A. Invertibility of DPM-Encoder

Proposition 1. *(Invertibility of DPM-Encoder) For each* $\boldsymbol{z} \sim \mathrm{DPMEnc}(\boldsymbol{z}|\boldsymbol{x}, G)$ *defined in Eq. (5), we have* $\boldsymbol{x} = \bar{\boldsymbol{x}} := G(\boldsymbol{z})$*, where* $\bar{\boldsymbol{x}} := G(\boldsymbol{z})$ *is defined as*

$$\begin{aligned} \bar{\boldsymbol{x}}_{T-1} &= \boldsymbol{\mu}_T(\boldsymbol{x}_T, T) + \boldsymbol{\sigma}_T \odot \boldsymbol{\epsilon}_T, \\ \bar{\boldsymbol{x}}_{t-1} &= \boldsymbol{\mu}_T(\bar{\boldsymbol{x}}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t, \quad T > t > 0, \\ \bar{\boldsymbol{x}} &:= \bar{\boldsymbol{x}}_0. \end{aligned} \tag{11}$$

Proof. We prove $\bar{\boldsymbol{x}}_t = \boldsymbol{x}_t$ for all $T - 1 \geq t \geq 0$ by induction. The proposition holds when $\bar{\boldsymbol{x}}_0 = \boldsymbol{x}_0$. To begin with, $\bar{\boldsymbol{x}}_{T-1} = \boldsymbol{x}_{T-1}$ because

$$\bar{\boldsymbol{x}}_{T-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_T, T) + \boldsymbol{\sigma}_T \odot \boldsymbol{\epsilon}_T \tag{12}$$

$$= \boldsymbol{\mu}_T(\boldsymbol{x}_T, T) + \boldsymbol{\sigma}_T \odot \big(\boldsymbol{x}_{T-1} - \boldsymbol{\mu}_T(\boldsymbol{x}_T, T)\big)/\boldsymbol{\sigma}_T = \boldsymbol{x}_{T-1}. \tag{13}$$

For $T - 1 \geq t > 0$, when $\bar{\boldsymbol{x}}_t = \boldsymbol{x}_t$, we have

$$\bar{\boldsymbol{x}}_{t-1} = \boldsymbol{\mu}_T(\bar{\boldsymbol{x}}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t \tag{14}$$

$$= \boldsymbol{\mu}_T(\boldsymbol{x}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t \tag{15}$$

$$= \boldsymbol{\mu}_T(\boldsymbol{x}_t, t) + \boldsymbol{\sigma}_t \odot \big(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_T(\boldsymbol{x}_t, t)\big)/\boldsymbol{\sigma}_t = \boldsymbol{x}_{t-1}. \tag{16}$$

□

B. Experimental Details of Zero-Shot Image Editing

Sources of images in the 150 tuples: For the zero-shot image-to-image translation experiment, we created a set of 150 tuples as task input, which include: (1) image generated by DALL·E 2 [24], (2) real images from [26], (3) real images from [7], (4) real images collected by the authors.

Per sample selection criterion: For each test sample, we allow each method to enumerate some combinations of hyperparameters (detailed below). To select the best combination for each sample, we used the directional CLIP score $\mathcal{S}_{\text{D-CLIP}}$ as the criterion (higher is better).

DDIB: DDIB edits images by using a deterministic DPM conditioned on the source text $\boldsymbol{t}$ to encode the source image, followed by decoding conditioned on the target text $\hat{\boldsymbol{t}}$. We used the deterministic DDIM sampler with 100 steps. We set the classifier-free guidance of the encoding step as 1; we enumerated the classifier-free guidance of the decoding step as $\{1, 1.5, 2, 3, 4, 5\}$.

SDEdit: SDEdit edits images by adding noise to the original image (the encoding step), followed by denoising the noised image with a diffusion model trained on the target domain (the decoding step). For zero-shot image-to-image translation, the decoding step of SDEdit uses the text-to-image diffusion model conditioned on the target image $\hat{\boldsymbol{t}}$. Notably, SDEdit does not provide a way to take the source text $\boldsymbol{t}$ as input. We used the DDIM sampler ($\eta = 0.1$) with 100 steps. We enumerated the classifier-free guidance of the decoding step as $\{1, 1.5, 2, 3, 4, 5\}$; we enumerated the encoding step as $\{15, 20, 25, 30, 40, 50\}$; we ran 15 trials for each hyperparameter combination.

CycleDiffusion: For our CycleDiffusion, we used the DDIM sampler ($\eta = 0.1$) with 100 steps. We set the classifier-free guidance of the encoding process as 1; we enumerated the classifier-free guidance of the decoding step as $\{1, 1.5, 2, 3, 4, 5\}$; we enumerated the early stopping step T_{es} as $\{15, 20, 25, 30, 40, 50\}$; we ran 15 trials for each hyperparameter combination.