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

(1) Unpaired image-to-image translation

(2) Zero-shot image editing

(3) Plug-and-play guidance

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

¹The 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]
requires 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 probabilistic 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 $\mu_T$ (usually parameterized as a UNet), the image $x := x_0$ is generated through the following process:

$$x_T \sim \mathcal{N}(0, I),$$
$$x_{t-1} = \mu_{T}(x_t, t) + \sigma_t \odot \epsilon_t, \quad t = T, \ldots, 1.$$  \hspace{1cm} (1)

Deterministic DPMs [29, 31, 11] generate images with the ODE formulation, which remove the randomness during the denoising process. Given the estimator $\mu_T$ (usually parameterized as a UNet) for the denoising direction, deterministic DPMs generate $x := x_0$ via

$$x_T \sim \mathcal{N}(0, I),$$
$$x_{t-1} = \mu_{T}(x_t, t), \quad t = T, \ldots, 1.$$  \hspace{1cm} (2)

3.2. Gaussian Latent Space for Diffusion Models

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

$$z := x_T \sim \mathcal{N}(0, I),$$
$$x_{t-1} = \mu_{T}(x_t, t), \quad t = T, \ldots, 1.$$  \hspace{1cm} (3)

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

$$z := (x_T \oplus \epsilon_T \oplus \cdots \oplus \epsilon_1) \sim \mathcal{N}(0, I),$$
$$x_{t-1} = \mu_{T}(x_t, t) + \sigma_t \odot \epsilon_t, \quad t = T, \ldots, 1.$$  \hspace{1cm} (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 $z$ from the image $x$ 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 $z$ from the image $x$ using stochastic DPMs.

3.3. DPM-Encoder

To sample latent codes $z$ from the image $x$ 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 $x := x_0$, stochastic DPMs define a posterior distribution $q(x_{1:T} | x_0)$ [9, 29]. Based on $q(x_{1:T} | x_0)$ and Eq. (4), we can directly derive $z \sim \text{DPMEnc}(z | x, G)$ as (see details in Appendix A)

$$x_1, \ldots, x_{T-1}, x_T \sim q(x_{1:T} | x_0),$$
$$\epsilon_t = (x_{t-1} - \mu_{T}(x_t, t)) / \sigma_t, \quad t = T, \ldots, 1,$$
$$z := (x_T \oplus \epsilon_T \oplus \cdots \oplus \epsilon_2 \oplus \epsilon_1).$$  \hspace{1cm} (5)

A property of DPM-Encoder is perfect reconstruction, meaning that we have $x = G(z)$ for every $z \sim \text{Enc}(z | x, G)$. 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 $D_1$ and $D_2$, 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 $z$. 

![Figure 2: Both deterministic and stochastic DPMs can be formulated as deterministic maps from latent code $z$ to image $x$.](image-url)

7380
from the image $x$, and then use $z$ to generate a new image $\hat{x} = G(z)$. 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 $x \in D_1$, we use DPM-Encoder to encode it as $z$ and then decode it as $\hat{x} = G_2(z)$:

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

We can also apply CycleDiffusion to text-to-image diffusion models by defining $D_1$ and $D_2$ as image distributions conditioned on two texts. Let $G_t$ be a text-to-image diffusion model conditioned on text $t$. Given a source image $x$, the user writes two texts: a source text $t$ describing the source image $x$ and a target text $t$ describing the target image $\hat{x}$ 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)

$$z \sim \text{DPMEnc}(z|\ x, G_t), \quad \hat{x} = G_t(z).$$

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

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

**Input:** source image $x := x_0$; source text $t$; target text $t$; encoding step $T_{es} \leq T$

1. Sample noisy image $\hat{x}_{T_{es}} = x_{T_{es}} \sim q(x_{T_{es}}|x_0)$
2. $x_{t-1} \sim q(x_{t-1}|x_t, x_0)$
3. $\epsilon_t = (x_{t-1} - \mu_T(x_t, t|t))/\sigma_t$
4. $\hat{x}_{t-1} = \mu_T(\hat{x}_t, t|t) + \sigma_t \epsilon_t$

**Output:** $\hat{x} := \hat{x}_{0}$

Let $B_t$ be the upper bound of $\|x_t - \hat{x}_t\|_2$ at time step $t$ when the same latent code $z$ is used for sampling (i.e., $x_0 = G_t(z)$ and $\hat{x}_0 = G_t(z)$). We have $B_T = 0$ because $\|x_T - \hat{x}_T\|_2 = 0$, and $B_0$ is the upper bound for the generated images $\|x - \hat{x}\|_2$. The upper bound $B_t$ can be propagated through time, from $T$ to 0. Specifically, by combining the above two properties, we have

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

### 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 $C$, one can define the guided image distribution as an energy-based model (EBM): $p(x|C) \propto p_x(x) e^{-\lambda_C E(x|C)}$. Sampling for $x \sim p(x|C)$ is equivalent to $z \sim p_z(z|C), x = G(z)$, where $p(z|C) \propto p_z(z) e^{-\lambda_C E(G(z|C))}$. Examples of the energy function $E(x|C)$ are provided in Section 4.3. To sample $z \sim p(z|C)$, one can use any model-agnostic samplers. For example, Langevin dynamics [34] starts from $z^{(0)} \sim \mathcal{N}(0, I)$ and samples $z := z^{(n)}$ iteratively through

$$z^{(k+1)} = z^{(k)} + \sigma/2 \nabla_z \left( \log p_z(z^{(k)}) - E(G(z^{(k)}|C)) \right) + \sqrt{\sigma} \omega^{(k)},$$

$$\omega^{(k)} \sim \mathcal{N}(0, I).$$

### 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 $256 \times 256$ for Cat → Dog translation and Wild → Dog translation. For each source image, each method should generate a target image with minimal changes. After CycleDiffusion generates the translated image, we used $T_{\text{sdedit}}$ steps of SDEdit to clean the artifacts. For $T = 1000$, we set $T_{\text{sdedit}} = 100$ for Cat → Dog and $T_{\text{sdedit}} = 125$ for Wild → 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
<table>
<thead>
<tr>
<th>Method</th>
<th>Cat → Dog</th>
<th>Wild → Dog</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FID↓</td>
<td>KID↑ x 10^3↓</td>
</tr>
<tr>
<td>CUT (GAN SOTA [21])</td>
<td>76.21</td>
<td>–</td>
</tr>
<tr>
<td>ILVR [3]</td>
<td>74.37</td>
<td>–</td>
</tr>
<tr>
<td>SDEdit [17]</td>
<td>74.17</td>
<td>–</td>
</tr>
<tr>
<td>EGSDE [37]</td>
<td>65.82</td>
<td>–</td>
</tr>
<tr>
<td>CycleDiffusion w/ DDIM (τ₁ = 0.1)</td>
<td><strong>58.87</strong></td>
<td>20.3</td>
</tr>
</tbody>
</table>

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 L₂ and PSNR [33].
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 $S_{D-CLIP}$. For a fair comparison, different methods share the same set of combinations of hyperparameters if possible, detailed in Appendix B.

Here is Table 2 that illustrates the zero-shot image editing results in a tabular format:

<table>
<thead>
<tr>
<th>Method</th>
<th>$S_{CLIP}↑$</th>
<th>$S_{D-CLIP}↑$</th>
<th>PSNR↑</th>
<th>SSIM↑</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LDM-400M</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDEdit [17]</td>
<td>0.332</td>
<td>0.264</td>
<td>13.68</td>
<td>0.390</td>
</tr>
<tr>
<td>DDIB [32]</td>
<td>0.324</td>
<td>0.195</td>
<td>15.82</td>
<td>0.544</td>
</tr>
<tr>
<td>CycleDiffusion w/ DDIM ($\eta = 0.1$; ours)</td>
<td>$\mathbf{0.333}$</td>
<td>$\mathbf{0.275}$</td>
<td>$\mathbf{18.72}$</td>
<td>$\mathbf{0.625}$</td>
</tr>
<tr>
<td><strong>Stable Diffusion v1-4</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDEdit [17]</td>
<td>$\mathbf{0.344}$</td>
<td>0.258</td>
<td>15.93</td>
<td>0.512</td>
</tr>
<tr>
<td>DDIB [32]</td>
<td>0.331</td>
<td>0.209</td>
<td>18.10</td>
<td>0.653</td>
</tr>
<tr>
<td>CycleDiffusion w/ DDIM ($\eta = 0.1$; ours)</td>
<td>0.334</td>
<td>$\mathbf{0.272}$</td>
<td>$\mathbf{21.92}$</td>
<td>$\mathbf{0.731}$</td>
</tr>
</tbody>
</table>

Figure 4: Examples of CycleDiffusion for zero-shot image editing. Source images $x$ are displayed with a purple margin; the others are images $\hat{x}$ 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:

$$S_{D-CLIP} = \cos \left( \text{CLIP}_{\text{img}}(\hat{x}) - \text{CLIP}_{\text{img}}(x), \right)$$

$$\text{CLIP}_{\text{text}}(\hat{t}) - \text{CLIP}_{\text{text}}(t) \right).$$ (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 baselines, 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., $S_{CLIP}$), 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.
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 $\mu_T(x_t, t | t)$ to $\mu_T(\hat{x}_t, t | \hat{t})$. However, we cannot apply it to all samples because CAC puts requirements on the difference between $t$ and $\hat{t}$. 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 $E_{\text{CLIP}}(x | t) = \frac{1}{L} \sum_{l=1}^{L} \left( 1 - \cos \langle \text{CLIP}_{\text{img}}(\text{DiffAug}_l(x)), \text{CLIP}_{\text{text}}(t) \rangle \right)$, 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


