# UNIFYING DIFFUSION MODELS' LATENT SPACE, WITH APPLICATIONS TO CYCLEDIFFUSION AND GUIDANCE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Diffusion models achieve unprecedented performance in image synthesis. Unlike GANs, VAEs, and normalizing flows, whose latent codes are Gaussian, the widely-adopted characterization of diffusion models' latent code is a sequence of gradually denoised images. This paper provides an alternative formulation of various diffusion models' latent space as Gaussian, along with an invertible **DPM-Encoder** that maps images into this latent space. While our formulation is directly derived from definition, we do demonstrate several intriguing consequences. (1) Empirically, we find that a common latent space emerges from two diffusion models trained independently on related domains. Based on this finding, we propose **CycleDiffusion**, which uses DPM-Encoder for unpaired image-to-image translation. (2) Applying CycleDiffusion to text-to-image diffusion models, we show that large-scale text-to-image diffusion models can be used as **zero-shot** image-to-image editors. (3) One can guide pre-trained diffusion models and GANs by controlling the latent codes in a unified, plug-and-play formulation, avoiding finetuning the guidance model on noisy images when guiding diffusion models. This property facilitates principled, fine-grained model comparisons on the same dataset. We demonstrate that, when guided by the CLIP model and a face recognition model, diffusion models have better coverage of low-density sub-populations and individuals than GANs.[1]

## 1 INTRODUCTION

Diffusion models (Song & Ermon, 2019; Ho et al., 2020) have achieved state-of-the-art performance and have been instrumental to revolutionary text-to-image models such as DALL·E 2 (Ramesh et al., 2022). Unlike GANs (Goodfellow et al., 2014), VAEs (Kingma & Welling, 2014), and normalizing flows (Dinh et al., 2015), whose latent codes are isometric Gaussian, the widely-adopted formulation of diffusion models' latent code is a sequence of images at different noise levels. This paper provides a unified view of generative models by reformulating various diffusion models as deterministic maps from Gaussian latent codes $\boldsymbol{z}$ to images $\boldsymbol{x}$ (Figure 1, Section 3.1). Once trained, the image distribution $\boldsymbol{x} \sim p_{\boldsymbol{x}}(\boldsymbol{x})$ is implicitly defined as $\underline{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z}), \boldsymbol{x} = G(\boldsymbol{z})}$.

A natural question following our unification is the *encoding* problem, i.e., how to embed an image $\boldsymbol{x}$ as a latent code $\boldsymbol{z}$. Encoding has been studied for many generative models. For instance, VAEs and normalizing flows have encoders by design; GAN inversion (Xia et al., 2021) learns a separate encoder after the GAN is trained; ODE-based deterministic diffusion models (Song et al., 2021a;b) build encoders with reverse ODEs. However, it still remains unclear what is the encoder for stochastic diffusion models. We propose **DPM-Encoder** (Section 3.2), an invertible "encoder" applicable to stochastic diffusion models such as DDPM (Ho et al., 2020), non-deterministic DDIM (Song et al., 2021a), and latent diffusion models (Rombach et al., 2022). Our unification and DPM-Encoder are purely from definition, but we demonstrate that several interesting consequences follow.

First, we observe that a common latent space naturally emerges from two diffusion models (with the same noise schedule but not necessarily the same architecture) trained independently on similar domains. Based on this finding, we propose **CycleDiffusion** (Section 3.3), a method for unpaired image-to-image translation based on DPM-Encoder. Like the GAN-based UNIT method (Liu et al., 2017), CycleDiffusion encodes and decodes images with the common latent space. Experiments show

[1]The code will be made publicly available.

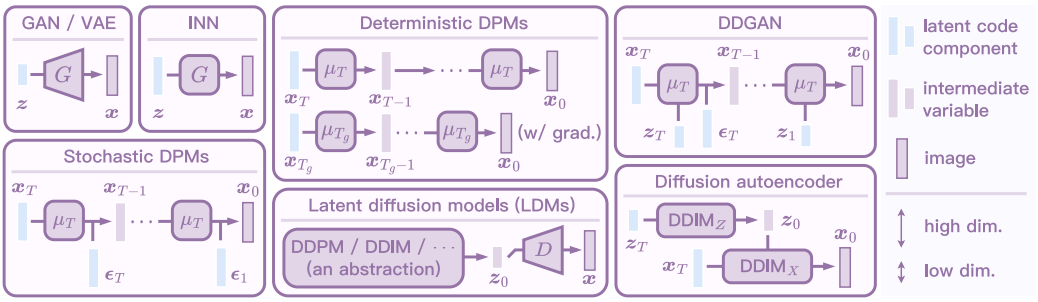



Figure 1: Once trained, various types of diffusion models can be reformulated as deterministic maps from latent codes $\boldsymbol{z}$ to images $\boldsymbol{x}$, like GANs, VAEs, and normalizing flows.

that CycleDiffusion outperforms previous methods based on GANs or diffusion models (Section 4.1). Furthermore, by applying CycleDiffusion to large-scale text-to-image diffusion models, we show that text-to-image diffusion models can be **zero-shot** image-to-image editors (Section 4.2).

Our unification also allows guiding diffusion models and GANs in a unified, plug-and-play formulation. Specifically, the guided image distribution $\boldsymbol{x} \sim p(\boldsymbol{x}|\mathcal{C})$ can be written as $\underline{\boldsymbol{z} \sim p(\boldsymbol{z}|\mathcal{C}), \boldsymbol{x} = G(\boldsymbol{z})}$ (Section 3.4). It avoids finetuning the guidance model on noisy images when guiding diffusion models (Dhariwal & Nichol, 2021; Liu et al., 2021). It facilitates principled, fine-grained model comparisons on the same data. With the CLIP model and a face embedding model as guidance, we show that diffusion models have a broader coverage of low-density sub-populations and individuals (Section 4.3).

## 2 Related Work

Recent years have witnessed a great progress in generative models such as GANs (Goodfellow et al., 2014), diffusion models (Song & Ermon, 2019; Ho et al., 2020; Dhariwal & Nichol, 2021), VAEs (Kingma & Welling, 2014), normalizing flows (Dinh et al., 2015), and their hybrid extensions (Sinha et al., 2021; Vahdat et al., 2021; Zhang & Chen, 2021; Kim et al., 2022a). Previous works have shown that their training objectives are related, e.g., diffusion models are essentially VAEs (Ho et al., 2020; Kingma et al., 2021; Huang et al., 2021); training objectives of GANs and VAEs have been unified as KL divergences (Hu et al., 2018) or mutual information with consistency constraints (Zhao et al., 2018); a recent attempt (Zhang et al., 2022b) has been made to unify generative models as GFlowNets (Bengio et al., 2021). Different from them, this work unifies generative models once they are trained. As the size of the training data grows, this unification can become increasingly useful.

One of the most fundamental challenges in generative modeling is to design an encoder that is both computationally efficient and invertible. GAN inversion learns an encoder for pre-trained GANs (Xia et al., 2021). VAEs and normalizing flows have their encoders by design. Song et al. (2021a;b) studied encoding for ODE-based deterministic diffusion models. However, it remains unclear how to encode for general stochastic diffusion models, and DPM-Encoder fills this blank. Also, CycleDiffusion can be seen as an extension of Su et al. (2022)'s DDIB approach to stochastic diffusion models.

Previous works have formulated plug-and-play guidance of generative models as latent-space EBMs (Nguyen et al., 2017; Nie et al., 2021; Wu et al., 2022), and our unification makes this formulation applicable to various diffusion models, which have been shown to be effective for modeling images, videos (Ho et al., 2022; Hoppe et al., 2022), molecules (Xu et al., 2022), 3D objects (Luo & Hu, 2021), and text (Li et al., 2022). This plug-and-play guidance can provide principled, fine-grained comparisons of generative models' coverage of sub-populations and individuals on the same dataset.

A concurrent work observed that fixing both (1) the random seed and (2) the cross attention map in Transformer-based text-to-image diffusion models results in images with minimal changes (Hertz et al., 2022); they leveraged this finding to edit *model-generated* images *when the random seed and the diffusion model (Imagen; Saharia et al., 2022) are known*. To edit *real* images, Hertz et al. (2022) further resorted to mask heuristics because "how to infer random seeds" for stochastic diffusion models was considered non-trivial. Notably, our DPM-Encoder is similar-in-spirit to inferring the "random seeds" because our $\boldsymbol{z}$ contains all randomness during sampling. Therefore, DPM-Encoder

can hopefully improve their performance on *real* image editing; however, since Hertz et al. (2022) and Imagen do not have open-sourced implementation, we leave this attempt for future work.

Table 1: Details of redefining various diffusion models' latent space (Section 3.1).

| | Latent code $\boldsymbol{z}$ | Deterministic map $\boldsymbol{x} = G(\boldsymbol{z})$ |
|---|---|---|
| DDPM *etc.* | $\boldsymbol{z} := (\boldsymbol{x}_T \oplus \boldsymbol{\epsilon}_T \oplus \cdots \oplus \boldsymbol{\epsilon}_1)$ | $\boldsymbol{x}_{T-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_T, T) + \boldsymbol{\sigma}_T \odot \boldsymbol{\epsilon}_T$, <br> $\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t \ (t < T)$, $\quad \boldsymbol{x} := \boldsymbol{x}_0$. |
| DDIM ($\eta = 0$) | $\boldsymbol{z} := \boldsymbol{x}_T$ ($T = T_g$ if with gradient) | $\boldsymbol{x}_{T-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_T, T)$, <br> $\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_t, t) \ (t < T)$, $\quad \boldsymbol{x} := \boldsymbol{x}_0$. |
| LDM | $\boldsymbol{z}$ of DDPM, DDIM, *etc.* | $\boldsymbol{z}_0 = G_{\text{latent}}(\boldsymbol{z})$, $\quad \boldsymbol{x} = D(\boldsymbol{z}_0)$. |
| DiffAE | $\boldsymbol{z} := (\boldsymbol{z}_T \oplus \boldsymbol{x}_T)$ | $\boldsymbol{z}_0 = \text{DDIM}_Z(\boldsymbol{z}_T)$, $\quad \boldsymbol{x} := \boldsymbol{x}_0 = \text{DDIM}_X(\boldsymbol{x}_T, \boldsymbol{z}_0)$. |
| DDGAN | $\boldsymbol{z} := (\boldsymbol{x}_T \oplus \boldsymbol{z}_T \oplus \boldsymbol{\epsilon}_T \oplus \cdots$ <br> $\oplus\, \boldsymbol{z}_2 \oplus \boldsymbol{\epsilon}_2 \oplus \boldsymbol{z}_1)$ | $\boldsymbol{x}_{T-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_T, \boldsymbol{z}_T, T) + \boldsymbol{\sigma}_T \odot \boldsymbol{\epsilon}_T$, <br> $\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_t, \boldsymbol{z}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t \ (1 < t < T)$, <br> $\boldsymbol{x} := \boldsymbol{x}_0 = \boldsymbol{\mu}_T(\boldsymbol{x}_1, \boldsymbol{z}_1, 1)$. |

# 3 METHOD

## 3.1 GAUSSIAN LATENT SPACE FOR DIFFUSION MODELS

Generative models such as GANs, VAEs, and normalizing flows can be seen as a family of *implicit models*, meaning that they are deterministic maps $G : \mathbb{R}^d \to \mathcal{X}$ from latent codes $\boldsymbol{z}$ to images $\boldsymbol{x}$. At inference, sampling from the image prior $\boldsymbol{x} \sim p_{\boldsymbol{x}}(\boldsymbol{x})$ is implicitly defined as $\underline{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z}), \boldsymbol{x} = G(\boldsymbol{z})}$. Generally, the latent prior $p_{\boldsymbol{z}}(\boldsymbol{z})$ is an isometric Gaussian distribution. In this section, we show how to unify diffusion models into this family. An overview is provided in Figure 1 and Table 1.

**Stochastic DPMs:** Stochastic DPMs (Ho et al., 2020; Song & Ermon, 2019; Song et al., 2021b;a; Watson et al., 2022) generate images with a Markov chain. Given the mean estimator $\boldsymbol{\mu}_T$ (see Appendix A) and $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, the image $\boldsymbol{x} := \boldsymbol{x}_0$ is generated through $\boldsymbol{x}_{t-1} \sim \mathcal{N}(\boldsymbol{\mu}_T(\boldsymbol{x}_t, t), \text{diag}(\boldsymbol{\sigma}_t^2))$. Using the reparameterization trick, we define the latent code $\boldsymbol{z}$ and the mapping $G$ recursively as

$$\boldsymbol{z} := (\boldsymbol{x}_T \oplus \boldsymbol{\epsilon}_T \oplus \cdots \oplus \boldsymbol{\epsilon}_1) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad \boldsymbol{x}_{t-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t, \quad t = T, \ldots, 1, \tag{1}$$

where $\oplus$ is concatenation. Here, $\boldsymbol{z}$ has dimension $d = d_I \times (T+1)$, where $d_I$ is the image dimension.

**Deterministic DPMs:** Deterministic DPMs (Song et al., 2021a;b; Salimans & Ho, 2022; Liu et al., 2022; Lu et al., 2022; Karras et al., 2022; Zhang & Chen, 2022) generates images with an ODE formulation. Given the mean estimator $\boldsymbol{\mu}_T$, deterministic DPMs generate $\boldsymbol{x} := \boldsymbol{x}_0$ via

$$\boldsymbol{z} := \boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad \boldsymbol{x}_{t-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_t, t), \quad t = T, \ldots, 1. \tag{2}$$

Since backpropagation through Eq. (2) is costly, we use fewer steps $T_g$ when computing gradients. Given the mean estimator $\boldsymbol{\mu}_{T_g}$ with discretization steps $T_g$, the image $\boldsymbol{x} := \boldsymbol{x}_0$ is generated as

$$\text{(with gradients)} \quad \boldsymbol{z} := \boldsymbol{x}_{T_g} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad \boldsymbol{x}_{t-1} = \boldsymbol{\mu}_{T_g}(\boldsymbol{x}_t, t), \quad t = T_g, \ldots, 1. \tag{3}$$

**Latent diffusion models (LDMs):** An LDM (Rombach et al., 2022) first uses a diffusion model $G_{\text{latent}}$ to compute a "latent code" $\boldsymbol{z}_0 = G_{\text{latent}}(\boldsymbol{z})$,[2] which is then decoded as $\boldsymbol{x} = D(\boldsymbol{z}_0)$. Note that $G_{\text{latent}}$ is an abstraction of the diffusion models that are already unified above.

**Diffusion autoencoder (DiffAE):** DiffAE (Preechakul et al., 2022) first uses a deterministic DDIM to generate a "latent code" $\boldsymbol{z}_0$,[2] which is used as condition for an image-space deterministic DDIM:

$$\boldsymbol{z} := (\boldsymbol{z}_T \oplus \boldsymbol{x}_T) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad \boldsymbol{z}_0 = \text{DDIM}_Z(\boldsymbol{z}_T), \quad \boldsymbol{x} := \boldsymbol{x}_0 = \text{DDIM}_X(\boldsymbol{x}_T, \boldsymbol{z}_0). \tag{4}$$

**DDGAN:** DDGAN (Xiao et al., 2022) models each reverse time step $t$ as a GAN conditioned on the previous step's output. We define DDGAN's latent code $\boldsymbol{z}$ and generation process $G$ as

$$\begin{aligned} &\boldsymbol{z} := (\boldsymbol{x}_T \oplus \boldsymbol{z}_T \oplus \boldsymbol{\epsilon}_T \oplus \cdots \oplus \boldsymbol{z}_2 \oplus \boldsymbol{\epsilon}_2 \oplus \boldsymbol{z}_1) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \\ &\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_t, \boldsymbol{z}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t, \quad t = T, \ldots, 2, \quad \boldsymbol{x} := \boldsymbol{x}_0 = \boldsymbol{\mu}_T(\boldsymbol{x}_1, \boldsymbol{z}_1, 1). \end{aligned} \tag{5}$$

[2]Quotation marks stand for "latent code" in the cited papers, different from our latent code $\boldsymbol{z}$ in Section 3.1.

Table 2: Encoding methods for generative models. †"Invertible" means that we have $\boldsymbol{x} = G(\boldsymbol{z})$ for every $\boldsymbol{z} \sim \mathrm{Enc}(\boldsymbol{z}|\boldsymbol{x}, G)$. ‡Depends on the errors caused by discretization.

| | 2D/3D GAN | VAE | Flow | DDIM ($\eta = 0$) | DDPM, DDIM ($\eta \neq \mathbf{0}$), *etc.* |
|---|---|---|---|---|---|
| Encoder | GAN inversion | encoder | $f^{-1}$ | DDIM ($\eta = 0$) encoder | DPM-Encoder (ours; Section 3.2) |
| Deterministic | ✗ | ✗ | ✓ | ✓ | ✗ |
| Invertible† | ✗ | ✗ | ✓ | ✓‡ | ✓ |

**Algorithm 1:** DPM-Encoder: an invertible "encoder" for stochastic DPMs

**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 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T-1}, \boldsymbol{x}_T \sim q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$
2. $\boldsymbol{z} = \boldsymbol{x}_T$
**for** $t = T, \ldots, 1$ **do**
  3. $\boldsymbol{\epsilon}_t = \big(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_T(\boldsymbol{x}_t, t)\big)/\boldsymbol{\sigma}_t$
  4. $\boldsymbol{z} = \boldsymbol{z} \oplus \boldsymbol{\epsilon}_t$

### 3.2 DPM-Encoder: an Invertible "Encoder" for Diffusion Models

In this section, we investigate the *encoding* problem, i.e., $\boldsymbol{z} \sim \mathrm{Enc}(\boldsymbol{z}|\boldsymbol{x}, G)$. Encoding methods have been studied for many generative models (Table 2), and our contribution is **DPM-Encoder**, an invertible "encoder" for stochastic DPMs. For each image $\boldsymbol{x} := \boldsymbol{x}_0$, stochastic DPMs define a posterior distribution $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ (Ho et al., 2020; Song et al., 2021a). Based on $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ and Eq. (1), we can directly derive $\boldsymbol{z} \sim \mathrm{DPMEnc}(\boldsymbol{z}|\boldsymbol{x}, G)$ as (see details in Appendices A and B)

$$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T-1}, \boldsymbol{x}_T \sim q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0), \quad \boldsymbol{\epsilon}_t = \big(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_T(\boldsymbol{x}_t, t)\big)/\boldsymbol{\sigma}_t, \quad t = T, \ldots, 1, \\ \boldsymbol{z} := \big(\boldsymbol{x}_T \oplus \boldsymbol{\epsilon}_T \oplus \cdots \oplus \boldsymbol{\epsilon}_2 \oplus \boldsymbol{\epsilon}_1\big). \tag{6}$$

DPM-Encoder also applies to latent diffusion models (LDMs). This is achieved by first using the LDM encoder $E$ to encode $\boldsymbol{z}_0 = E(\boldsymbol{x})$ and then sampling $\boldsymbol{z} \sim \mathrm{DPMEnc}(\boldsymbol{z}|\boldsymbol{z}_0, G_{\text{latent}})$.

### 3.3 CycleDiffusion: Image-to-Image Translation with DPM-Encoder

Suppose we have two stochastic DPMs $G_1$ and $G_2$ trained on similar domains $\mathcal{D}_1$ and $\mathcal{D}_2$ (e.g., cats and dogs) independently, and the two models have the same noise schedule $\boldsymbol{\sigma}_t$ but not necessarily the same architecture. We find that the two models naturally learn a common latent space. In other words, for each latent code $\boldsymbol{z}$, $G_1(\boldsymbol{z})$ and $G_2(\boldsymbol{z})$ have common details. One explanation is that, in Eq. (1), $\boldsymbol{\mu}_T(\boldsymbol{x}_t, t)$ models patterns (i.e., averaged direction pointing to the clean image) in one domain, while $\boldsymbol{\epsilon}_t$ models details that are specific to one image. Based on this finding, we propose **CycleDiffusion**, an unpaired image-to-image translation method using our DPM-Encoder. Given a source image $\boldsymbol{x} \in \mathcal{D}_1$, we encode it as $\boldsymbol{z}$ and then decode it as $\hat{\boldsymbol{x}} = G_2(\boldsymbol{z})$:

$$\boldsymbol{z} \sim \mathrm{DPMEnc}(\boldsymbol{z}|\boldsymbol{x}, G_1), \quad \hat{\boldsymbol{x}} = G_2(\boldsymbol{z}). \tag{7}$$

Domains $\mathcal{D}_1$ and $\mathcal{D}_2$ do not need to be two datasets. For instance, we can apply CycleDiffusion to text-to-image diffusion models, where $\mathcal{D}_1$ and $\mathcal{D}_2$ are 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. Notably, since the text-to-image diffusion model has never seen image-to-image pairs during pre-training, this setting is **zero-shot** image-to-image editing. Formally,

$$\boldsymbol{z} \sim \mathrm{DPMEnc}(\boldsymbol{z}|\boldsymbol{x}, G_{\boldsymbol{t}}), \quad \hat{\boldsymbol{x}} = G_{\hat{\boldsymbol{t}}}(\boldsymbol{z}). \tag{8}$$

Detailed algorithm descriptions can be found in Appendix C, where we also design different variants of CycleDiffusion, such as noise removal and early stopping, to further improve the performance.

### 3.4 Unified Plug-and-Play Guidance for Generative Models

Prior works showed that guidance for generative models can be achieved in the latent space (Nguyen et al., 2017; Nie et al., 2021; Wu et al., 2022). Given a condition $\mathcal{C}$, we follow Wu et al. (2022) to

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 $\underline{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z}|\mathcal{C}), \boldsymbol{x} = G(\boldsymbol{z})}$ (Wu et al., 2022), 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 (Welling & Teh, 2011) starts from $\boldsymbol{z}^{\langle 0 \rangle} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and samples $\boldsymbol{z} := \boldsymbol{z}^{\langle n \rangle}$ iteratively through

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

Table 3: 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 Zhao et al. (2022). Best results using diffusion models are in **bold**. CycleDiffusion has the best FID and KID 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 (Wang et al., 2004).

| | Cat → Dog | | | | Wild → Dog | | | |
|---|---|---|---|---|---|---|---|---|
| | FID↓ | KID×$10^3$↓ | PSNR↑ | SSIM↑ | FID↓ | KID×$10^3$↓ | PSNR↑ | SSIM↑ |
| CUT (GAN SOTA; Park et al., 2020) | 76.21 | – | 17.48 | 0.601 | 92.94 | – | 17.20 | 0.592 |
| ILVR (Choi et al., 2021) | 74.37 | – | 17.77 | 0.363 | 75.33 | – | 16.85 | 0.287 |
| SDEdit (Meng et al., 2022) | 74.17 | – | 19.19 | 0.423 | 68.51 | – | 17.98 | 0.343 |
| EGSDE (Zhao et al., 2022) | 65.82 | – | **19.31** | 0.415 | 59.75 | – | **18.14** | 0.343 |
| CycleDiffusion w/ DDIM ($\eta = 0.1$) | **55.93** | **18.6** | 16.78 | **0.537** | **54.26** | **16.9** | 16.47 | **0.449** |

# 4 Experiments

This section provides experimental validation of claims in the previous section. Section 4.1 shows that CycleDiffusion achieves competitive results on unpaired image-to-image translation benchmarks. Section 4.2 shows that applying CycleDiffusion to large-scale text-to-image diffusion models makes them zero-shot image-to-image editors. In Section 4.3, diffusion models and GANs are guided by the CLIP model and a face recognition model in a unified, plug-and-play formulation.

## 4.1 CycleDiffusion for Unpaired Image-to-Image Translation

Given two unaligned image domains, unpaired image-to-image translation aims at mapping images in one domain to the other. We follow setups from previous works whenever possible, as detailed below.

**Datasets:** Following previous work (Park et al., 2020; Zhao et al., 2022), we conducted experiments on the test set of AFHQ (Choi et al., 2020) with resolution $256 \times 256$ for Cat → Dog and Wild → Dog. We also experimented with FFHQ (Karras et al., 2019) → CelebAHQ (Karras et al., 2018); since the baselines did not have this translation direction, we only show samples in Figure 2.

**Baselines:** We compared CycleDiffusion with previous unpaired image-to-image translation methods: CUT (Park et al., 2020), ILVR (Choi et al., 2021), SDEdit (Meng et al., 2022), EGSDE (Zhao et al., 2022), and DDIB (Su et al., 2022). We implemented DDIB following the original paper, but we only manage to get it to work on the FFHQ → CelebAHQ direction, so we omit it in the main table.

**Metrics:** To evaluate the realism, we reported Frechet Inception Distance (FID; Heusel et al., 2017) and Kernel Inception Distance (KID; Bińkowski et al., 2018) between the set of translated images and the set of target images. To evaluate the faithfulness, we reported PSNR and SSIM (Wang et al., 2004) between each translated image and its source image, averaged across all instances.

**Pre-trained diffusion models:** ILVR, SDEdit, and EGSDE only need the diffusion model trained on the target domain (i.e., Dog). We followed them to use the pre-trained model from Choi et al. (2021) for Dog. DDIB and CycleDiffusion need diffusion models on both domains, so we trained the Cat and Wild models. We used the pre-trained FFHQ model from Choi et al. (2021) and CelebAHQ model from Meng et al. (2022); we also experimented with pre-trained LDMs (Rombach et al., 2022).

Seen in Table 3 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 (Wang et al., 2004). Among all diffusion model-based methods, CycleDiffusion achieves the highest SSIM. Figure 2 displays some image samples. Since our implementation of DDIB has severe degeneration on AFHQ, we only present it on the FFHQ → CelebAHQ direction. One possible reason is diffusion models trained on Cat and Wild have lower quality as we observed. Moreover, results with LDM-DDIMs are consistent with those with DDIMs.

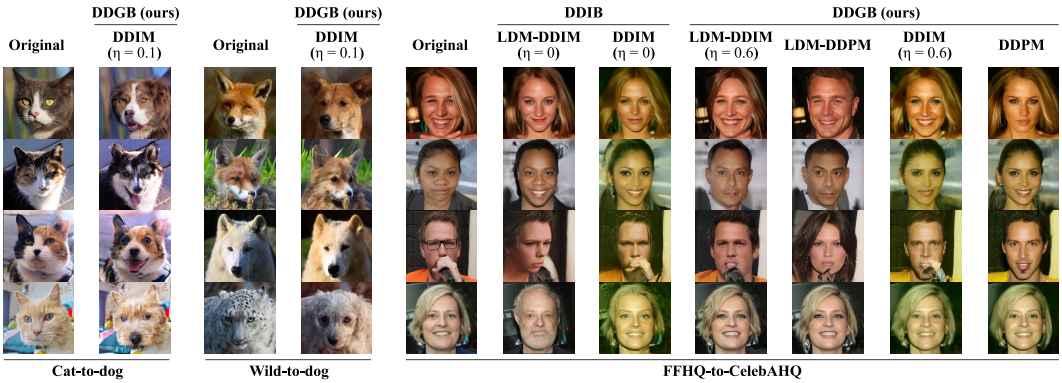



Figure 2: Unpaired image-to-image translation (Cat → Dog, Wild → Dog, and FFHQ → CelebAHQ).

### 4.2 Text-to-Image Diffusion Models Can Be Zero-Shot Image-to-Image Editors

This section shows that text-to-image diffusion models can be used for **zero-shot** image-to-image translation. Following our notation in Section 3.3, the input is a triple $(\boldsymbol{x}, \boldsymbol{t}, \hat{\boldsymbol{t}})$, where $\boldsymbol{x}$ is the source image, $\boldsymbol{t}$ is the source text, and $\hat{\boldsymbol{t}}$ is the target text. We denote the target image as $\hat{\boldsymbol{x}}$. There are no available data or evaluation metrics for this task, and we provide our experimental setup below.

**Dataset:** We curated a set of 150 triples $(\boldsymbol{x}, \boldsymbol{t}, \hat{\boldsymbol{t}})$ for this task. Notably, Hertz et al. (2022) provided several triples $(\boldsymbol{x}, \boldsymbol{t}, \hat{\boldsymbol{t}})$, and we included some of them; however, it is not fair to compare with their results when $\boldsymbol{x}$ is generated by Imagen (Saharia et al., 2022) since Hertz et al. (2022) used Imagen as a white-box and assumed that the random seed used for generation is known.

**Baselines:** The baselines include SDEdit (Meng et al., 2022) and DDIB (Su et al., 2022). 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 D).

**Metrics:** To evaluate the faithfulness to source images, we reported PSNR and SSIM. To evaluate the authenticity to the target text, we reported the CLIP score (CLIP embeddings are normalized): $\mathcal{S}_{\text{CLIP}}(\hat{\boldsymbol{x}}|\hat{\boldsymbol{t}}) = \cos\left\langle \text{CLIP}_{\text{img}}(\hat{\boldsymbol{x}}), \text{CLIP}_{\text{text}}(\hat{\boldsymbol{t}})\right\rangle$. We note a trade-off between PSNR/SSIM and $\mathcal{S}_{\text{CLIP}}$: by copying the source image we get high PSNR/SSIM, and by ignoring the source image we get high $\mathcal{S}_{\text{CLIP}}$. Here, we try to bypass this trade-off by reporting the directional CLIP score (Patashnik et al., 2021) (both CLIP embeddings and the embedding differences are normalized):

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

**Pre-trained text-to-image diffusion models:** To investigate how the zero-shot performance changes with data size, data quality, and training details, we used the following models: (1) `LDM-400M`, a 1.45B-parameter model trained on LAION-400M (Schuhmann et al., 2021), (2) `SD-v1-1`, a 0.98B-parameter Stable Diffusion model trained on LAION-5B (Schuhmann et al., 2022), (3) `SD-v1-4`, finetuned from `SD-v1-1` for improved aesthetics and classifier-free guidance.

Seen in Table 4 are results for zero-shot image-to-image translation. When the same pre-trained text-to-image model is used, CycleDiffusion achieves the best zero-shot performance. 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 far away from the source images (see Appendix F). For all methods, using `SD-v1-1` or `SD-v1-4` as the pre-trained text-to-image diffusion model results in better faithfulness than using `LDM-400M`. Figure 3 provides several samples from our CycleDiffusion, demonstrating that the edits are minimal and meaningful.

Table 4: Quantitative evaluation for **zero-shot** image-to-image translation.

| | Method | $\mathcal{S}_{\text{CLIP}}\uparrow$ | $\mathcal{S}_{\text{D-CLIP}}\uparrow$ | PSNR$\uparrow$ | SSIM$\uparrow$ |
|---|---|---|---|---|---|
| LDM-400M | SDEdit (Meng et al., 2022) | 0.332 | 0.264 | 13.68 | 0.390 |
| | DDIB (Su et al., 2022) | 0.324 | 0.195 | 15.82 | 0.544 |
| | CycleDiffusion w/ DDIM ($\eta = 0.1$; ours) | **0.333** | **0.275** | **18.72** | **0.625** |
| SD-v1-1 | SDEdit (Meng et al., 2022) | **0.339** | 0.248 | 15.50 | 0.498 |
| | DDIB (Su et al., 2022) | 0.324 | 0.189 | 18.58 | 0.658 |
| | CycleDiffusion w/ DDIM ($\eta = 0.1$; ours) | 0.331 | **0.262** | **21.98** | **0.731** |
| SD-v1-4 | SDEdit (Meng et al., 2022) | **0.344** | 0.258 | 15.93 | 0.512 |
| | DDIB (Su et al., 2022) | 0.331 | 0.209 | 18.10 | 0.653 |
| | CycleDiffusion w/ DDIM ($\eta = 0.1$; ours) | 0.334 | **0.272** | **21.92** | **0.731** |

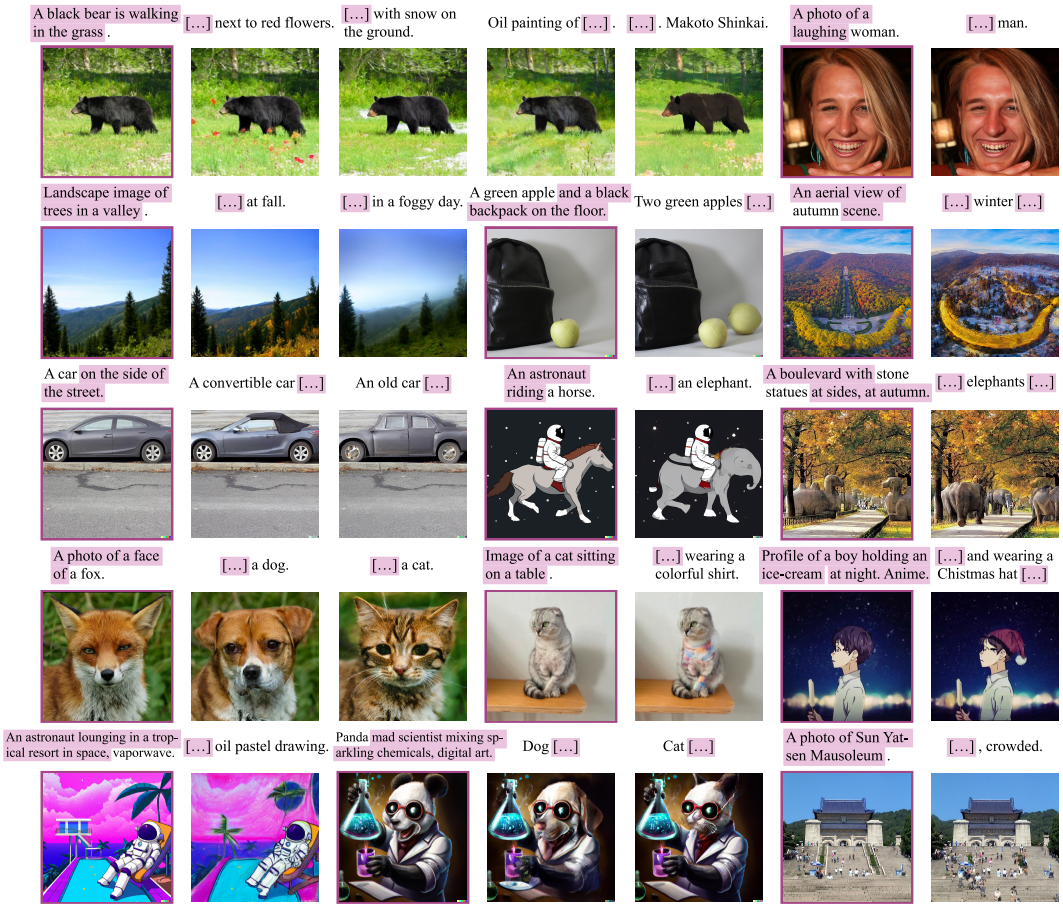
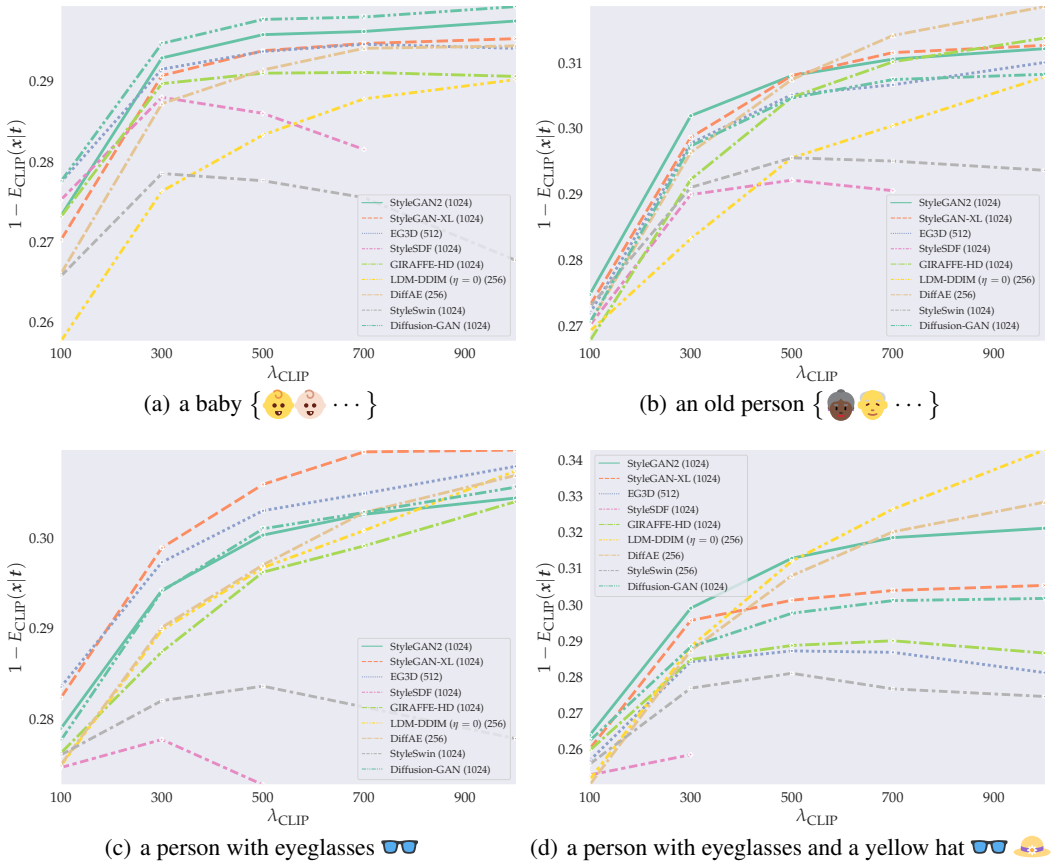



Figure 3: Text-to-image diffusion models can be zero-shot image-to-image editors. Source images $\boldsymbol{x}$ are displayed with a purple margin; the others are generated target images $\hat{\boldsymbol{x}}$. Overlapping text spans are marked in purple in source texts $\boldsymbol{t}$ and abbreviated as $[\ldots]$ in target texts $\hat{\boldsymbol{t}}$.

### 4.3 Unified Plug-and-Play Guidance for Diffusion Models and GANs

Existing works on guiding diffusion models need to train or finetune the guidance model on noisy images (Dhariwal & Nichol, 2021; Liu et al., 2021), which deviates from the idea of plug-and-play guidance of generative models (Nguyen et al., 2017). Our unification allows for unified plug-and-play guidance of diffusion models and GANs. It facilitates principled comparisons over sub-populations and individuals when the models are trained on the same dataset and guided by the same off-the-shelf model. For sub-population, we follow Wu et al. (2022) to use the CLIP model and text $\boldsymbol{t}$ as guidance and define the energy in Section 3.4 as $E_{\text{CLIP}}(\boldsymbol{x}|\boldsymbol{t}) = \frac{1}{L}\sum_{l=1}^{L}\left(1 - \cos\left\langle \text{CLIP}_{\text{img}}\left(\text{DiffAug}_l(\boldsymbol{x})\right), \text{CLIP}_{\text{text}}(\boldsymbol{t})\right\rangle\right)$




(a) a baby { ⋯ }

(b) an old person { ⋯ }

(c) a person with eyeglasses

(d) a person with eyeglasses and a yellow hat

Figure 4: Unified plug-and-play guidance for diffusion models and GANs with text and CLIP. The text description used in each plot is a photo of [____]. Image samples and more analyses are in Figure 5 and Appendix H. When the guidance becomes complex, diffusion models surpass GANs.

where each $\text{DiffAug}_l$ is differentiable augmentation (Zhao et al., 2020). We enumerated the guidance strength $\lambda_{\text{CLIP}} \in \{100, 300, 500, 700, 1000\}$. For sampling, we used Langevin dynamics in Eq. (9) with $n = 200$, $\sigma = 0.05$. For evaluation, we reported $(1 - E_{\text{CLIP}}(\boldsymbol{x}|\boldsymbol{t}))$ averaged over 256 samples.

Figure 4 plots models with pre-trained weights on FFHQ (Karras et al., 2019) (citations in Table 6). In Figure 5, we also visualize samples for SN-DDPM and DDGAN trained on CelebA. Our findings are as follows. (1) Diffusion models enable complex controls, e.g., when the text is (Figures 4-5). (2) 3D GANs are less controllable than 2D GANs and diffusion models. (3) Some models have undesirable generalizations, e.g., generating baby-like adults instead of babies (Figure 5). (4) For the same model, non-truncated (Karras et al., 2019) and higher-resolution variants are more controllable (Tables 7-8). (5) Models learn different biases of the same sub-population, e.g., Figures 5(g), 5(h), and 5(i) show that, compared to DiffAE and Diffusion-StyleGAN2, EG3D is less biased for gender.

Broad coverage of individuals is an important aspect of the personalized use of generative models. To analyze this coverage, we control different models to generate images that are close to a reference $\boldsymbol{x}_r$ in the identity (ID) space. Specifically, we used the IR-SE50 face embedding model (Deng et al., 2019), denoted as $R$, and defined $E_{\text{ID}}(\boldsymbol{x}|\boldsymbol{x}_r) = 1 - \cos\langle R(\boldsymbol{x}), R(\boldsymbol{x}_r)\rangle$ with strength $\lambda_{\text{ID}} = 2500$. For sampling, we used Langevin dynamics in Eq. (9) with $n = 200$ and $\sigma = 0.05$. To measure ID similarity to $\boldsymbol{x}_r$, we report $\cos\langle R(\boldsymbol{x}), R(\boldsymbol{x}_r)\rangle$, averaged over 256 samples.

In Table 5, we report the performance of StyleGAN2, StyleGAN-XL, GIRAFFE-HD, EG3D, LDM-DDIM, DDGAN, and DiffAE. DDGAN is trained on CelebAHQ, while others are trained on FFHQ. We find that diffusion models have much better coverage of individuals than 2D/3D GANs. Among diffusion models, deterministic LDM-DDIM ($\eta = 0$) achieves the best identity guidance performance.

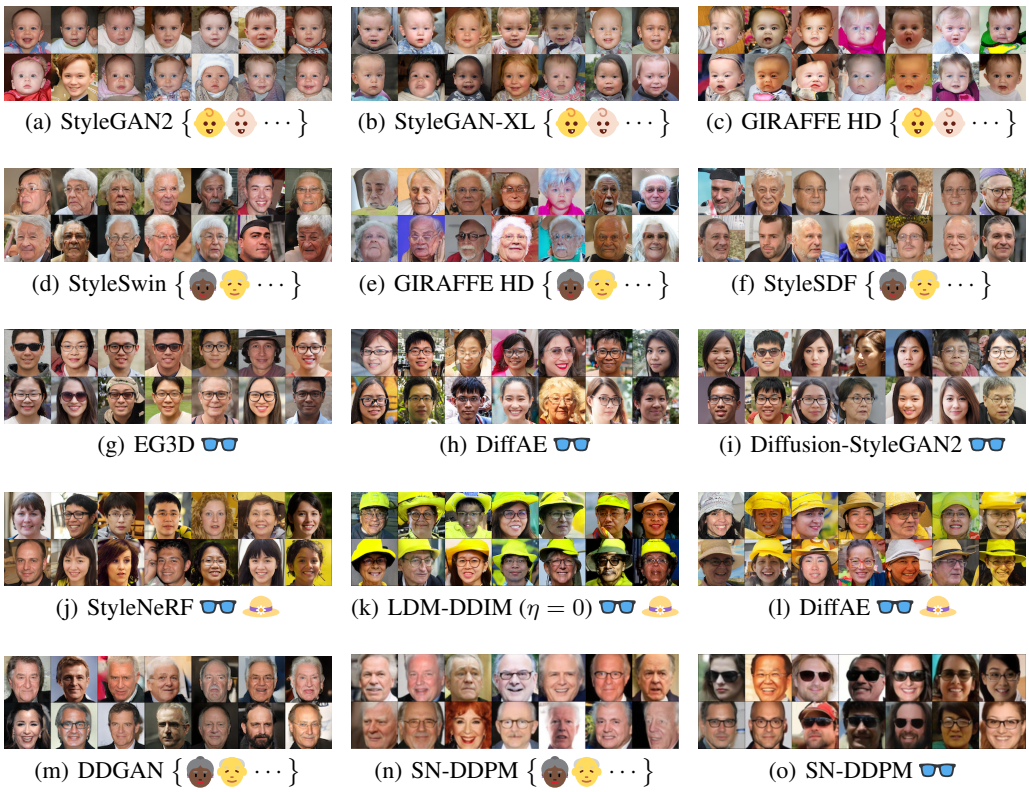



Figure 5: Sampling sub-populations from pre-trained generative models. Notations follow Figure 4. Images in (a)-(l) are from models trained on FFHQ, guided by the CLIP model and text; images in (m)-(o) are from models trained on CelebA(HQ), guided by attribute classifiers trained on CelebA.

Table 5: Guiding diffusion models and GANs with ID. ID-{A, B, C, D} are images from FFHQ. The metric is the ArcFace cosine similarity (Deng et al., 2019). See samples in Figure 8 (Appendix H).

| | *2D GAN* | | *3D GAN* | | *Diffusion model* | | |
|---|---|---|---|---|---|---|---|
| | StyleGAN2 | StyleGAN-XL | GIRAFFE-HD | EG3D | LDM-DDIM ($\eta = 0$) | DDGAN | DiffAE |
| ID-A | 0.561 | 0.681 | 0.616 | 0.468 | **0.904** | 0.837 | 0.873 |
| ID-B | 0.604 | 0.688 | 0.590 | 0.454 | **0.896** | 0.805 | 0.838 |
| ID-C | 0.495 | 0.636 | 0.457 | 0.403 | **0.892** | 0.795 | 0.852 |
| ID-D | 0.554 | 0.687 | 0.574 | 0.436 | **0.911** | 0.831 | 0.873 |

## 5 Conclusions and Discussion

This paper provides a unified view of pre-trained deep generative models by reformulating the latent space for different diffusion models. Although this reformulation is purely derived from definition, we showed that it has made diffusion models behave more like CycleGANs (see our CycleDiffusion) and GANs (see our plug-and-play guidance). Our definition of diffusion models' latent code $z$ also reinforces the importance of learnable covariance matrices in the reserve-time process, which are directly multiplied with $z$, in the reverse process (Bao et al., 2022a;b). Besides the interesting results, it is worth noting that this paper raises more questions than it answers. First of all, the mechanism behind the common latent space for stochastic DPMs, which is the basis of our CycleDiffusion, needs further exploration. Notably, Khrulkov & Oseledets (2022) and Su et al. (2022) provided explanations in the case of deterministic DPMs based on the optimal transport, while it is unclear whether it can be adapted to stochastic DPMs. Moreover, for stochastic DPMs on high-resolution images with many diffusion steps, how to enable efficient plug-and-play guidance still remains an open question.

## REFERENCES


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

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *ICLR*, 2022b.

Yoshua Bengio, Tristan Deleu, Edward J. Hu, Salem Lahlou, Mo Tiwari, and Emmanuel Bengio. GFlowNet foundations. *ArXiv*, 2021.

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

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *ICLR*, 2019.

Eric Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, S. Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. *CVPR*, 2022.

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

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

Jiankang Deng, J. Guo, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. *CVPR*, 2019.

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

Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. *ICLR, Workshop Track Proceedings*, 2015.

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.

Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleNeRF: A style-based 3D aware generator for high-resolution image synthesis. *ICLR*, 2022.

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.

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.

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

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *NeurIPS*, 2022.

Tobias Hoppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *NeurIPS*, 2022.

Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric P. Xing. On unifying deep generative models. *ICLR*, 2018.

Chin-Wei Huang, Jae Hyun Lim, and Aaron C. Courville. A variational perspective on diffusion-based generative models and score matching. *NeurIPS*, 2021.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *ICLR*, 2018.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CVPR*, 2019.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. *CVPR*, 2020.

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

Valentin Khrulkov and I. Oseledets. Understanding DDPM latent codes through optimal transport. *ArXiv*, 2022.

Dongjun Kim, Byeonghu Na, Se Jung Kwon, Dongsoo Lee, Wanmo Kang, and Il-Chul Moon. Maximum likelihood training of implicit nonlinear diffusion models. *NeurIPS*, 2022a.

Gwanghyun Kim, Taesung Kwon, and Jong-Chul Ye. DiffusionCLIP: Text-guided diffusion models for robust image manipulation. *CVPR*, 2022b.

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

Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *NeurIPS*, 2021.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. Diffusion-LM improves controllable text generation. *NeurIPS*, 2022.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *ICLR*, 2022.

Ming-Yu Liu, Thomas M. Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *NIPS*, 2017.

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.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *ICCV*, 2015.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. *NeurIPS*, 2022.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. *CVPR*, 2021.

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.

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.

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

Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. StyleSDF: High-resolution 3D-consistent image and geometry generation. *CVPR*, 2022.

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

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

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

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

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

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *ICLR*, 2022.

Axel Sauer, Katja Schwarz, and Andreas Geiger. StyleGAN-XL: Scaling StyleGAN to large diverse datasets. *SIGGRAPH*, 2022.

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.

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.

Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. InterFaceGAN: Interpreting the disentangled face representation learned by GANs. *TPAMI*, 2022.

Abhishek Sinha, Jiaming Song, Chenlin Meng, and Stefano Ermon. D2C: Diffusion-denoising models for few-shot conditional generation. *NeurIPS*, 2021.

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

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

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

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

Cristian Vaccari and Andrew Chadwick. Deepfakes and disinformation: Exploring the impact of synthetic political video on deception, uncertainty, and trust in news. *Social Media + Society*, 6, 2020.

Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. *NeurIPS*, 2020.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *NeurIPS*, 2021.

Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-GAN: Training GANs with diffusion. *ArXiv*, 2022.

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.

Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. *ICLR*, 2022.

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

Mika Westerlund. The emergence of Deepfake technology: A review. *Technology Innovation Management Review*, 9(11), 2019.

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.

Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. GAN inversion: A survey. *ArXiv*, 2021.

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

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *ICLR*, 2022.

Yang Xue, Yuheng Li, Krishna Kumar Singh, and Yong Jae Lee. GIRAFFE HD: A high-resolution 3D-aware generative model. *CVPR*, 2022.

Bo Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. StyleSwin: Transformer-based GAN for high-resolution image generation. *CVPR*, 2022a.

Dinghuai Zhang, Ricky T. Q. Chen, Nikolay Malkin, and Yoshua Bengio. Unifying generative models with GFlowNets. *ArXiv*, 2022b.

Qinsheng Zhang and Yongxin Chen. Diffusion normalizing flow. *NeurIPS*, 2021.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *ArXiv*, 2022.

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

Shengjia Zhao, Jiaming Song, and Stefano Ermon. The information-autoencoding family: A Lagrangian perspective on latent variable generative modeling. *ArXiv*, 2018.

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