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

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

ABSTRACT

Diffusion models have achieved unprecedented performance in generative modeling. The commonly-adopted formulation of the latent code of diffusion models is a sequence of gradually denoised samples, as opposed to the simpler (e.g., Gaussian) latent space of GANs, VAEs, and normalizing flows. This paper provides an alternative, Gaussian formulation of the latent space of diffusion models, as well as a reconstructable **DPM-Encoder** that maps images into the latent space. While our formulation is purely based on the definition of diffusion models, we demonstrate several intriguing consequences. (1) Empirically, we observe that a common latent space emerges from two diffusion models trained independently on related domains. In light of this finding, we propose **CycleDiffusion**, which uses DPM-Encoder for unpaired image-to-image translation. Furthermore, 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. (2) One can guide pre-trained diffusion models and GANs by controlling the latent codes in a unified, plug-and-play formulation based on energy-based models. Using the CLIP model and a face recognition model as guidance, we demonstrate that 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 unprecedented results in generative modeling and are instrumental to 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), which have a simple (e.g., Gaussian) latent space, the commonly-adopted formulation of the "latent code" of diffusion models is a sequence of gradually denoised images. This formulation makes the prior distribution of the "latent code" data-dependent, deviating from the idea that generative models are mappings from simple noises to data (Goodfellow et al., 2014).

This paper provides a unified view of generative models of images by reformulating various diffusion models as deterministic maps from a Gaussian latent code $\boldsymbol{z}$ to an image $\boldsymbol{x}$ (Figure 1, Section 3.1). A question that follows is *encoding*: how to map an image $\boldsymbol{x}$ to 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) builds *post hoc* encoders for GANs, and deterministic diffusion probabilistic models (DPMs) (Song et al., 2021a;b) build encoders with forward ODEs. However, it is still unclear how to build an encoder for stochastic DPMs such as DDPM (Ho et al., 2020), non-deterministic DDIM (Song et al., 2021a), and latent diffusion models (Rombach et al., 2022). We propose **DPM-Encoder** (Section 3.2), a reconstructable encoder for stochastic DPMs.

We show that some intriguing consequences emerge from our definition of the latent space of diffusion models and our DPM-Encoder. First, observations have been made that, given two diffusion models, a fixed "random seed" produces similar images (Nichol et al., 2022). Under our formulation, we formalize "similar images" via an upper bound of image distances. Since the defined latent code contains all randomness during sampling, DPM-Encoder is similar-in-spirit to inferring the "random seed" from real images. Based on this intuition and the upper bound of image distances,

[1]The code is publicly available at https://github.com/ChenWu98/cycle-diffusion.

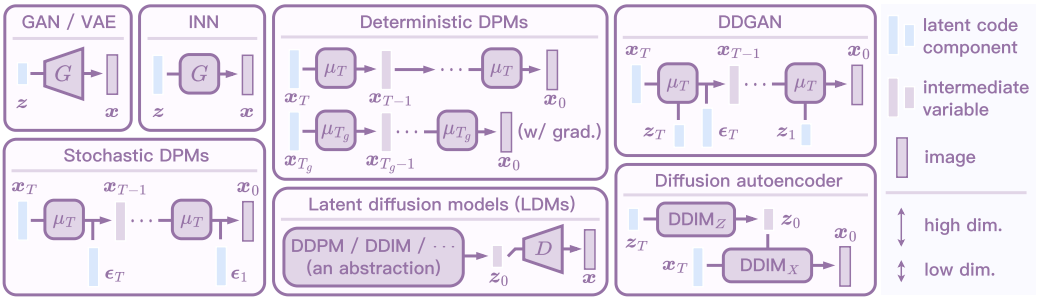


Figure 1: Once trained, various types of diffusion models can be reformulated as deterministic maps from latent code $\boldsymbol{z}$ to image $\boldsymbol{x}$, like GANs, VAEs, and normalizing flows.

we propose **CycleDiffusion** (Section 3.3), a method for unpaired image-to-image translation using our DPM-Encoder. Like the GAN-based UNIT method (Liu et al., 2017), CycleDiffusion encodes and decodes images using the common latent space. Our experiments show that CycleDiffusion outperforms previous methods based on GANs or diffusion models (Section 4.1). Furthermore, by applying large-scale text-to-image diffusion models (e.g., Stable Diffusion; Rombach et al., 2022) to CycleDiffusion, we obtain **zero-shot** image-to-image editors (Section 4.2).

With a simple latent prior, generative models can be guided in a plug-and-play manner by means of energy-based models (Nguyen et al., 2017; Nie et al., 2021; Wu et al., 2022). Thus, our unification allows unified, plug-and-play guidance for various diffusion models and GANs (Section 3.4), which avoids finetuning the guidance model on noisy images for diffusion models (Dhariwal & Nichol, 2021; Liu et al., 2021). With the CLIP model and a face recognition model as guidance, we show that diffusion models have 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 as VAEs (Ho et al., 2020; Kingma et al., 2021; Huang et al., 2021); GANs and VAEs 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 several generative models as GFlowNets (Bengio et al., 2021). In contrast, this paper unifies generative models as deterministic mappings from Gaussian noises to data (*aka* implicit models) once they are trained. Generative models with non-Gaussian randomness (Davidson et al., 2018; Nachmani et al., 2021) can be unified as deterministic mappings in similar ways.

One of the most fundamental challenges in generative modeling is to design an encoder that is both computationally efficient and invertible. GAN inversion trains an encoder after GANs are pre-trained (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 probabilistic models (DPMs). However, it remains unclear how to encode for general stochastic DPMs, and DPM-Encoder fills this gap. Also, CycleDiffusion can be seen as an extension of Su et al. (2022)'s DDIB approach to stochastic DPMs.

Previous works have formulated plug-and-play guidance of generative models as latent-space energy-based models (EBMs) (Nguyen et al., 2017; Nie et al., 2021; Wu et al., 2022), and our unification makes it applicable to various diffusion models, which are effective for modeling images, audio (Kong et al., 2021), 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 model comparisons of 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). The idea of fixing the cross-attention map is named Cross Attention Control (CAC)

in that work, which can be used to edit *model-generated* images *when the random seed is known*. For *real* images with stochastic DPMs, they generate masks based on the attention map because the random seed is unknown for real images. In Section 4.2, we show that CycleDiffusion and CAC can be combined to improve the structural preservation of image editing.

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

	Latent code $\boldsymbol{z}$	Deterministic map $\boldsymbol{x} = G(\boldsymbol{z})$
Stochastic DPMs	$\boldsymbol{z} := (\boldsymbol{x}_T \oplus \boldsymbol{\epsilon}_T \oplus \cdots \oplus \boldsymbol{\epsilon}_1)$	$\boldsymbol{x}_{T-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_T, T) + \boldsymbol{\sigma}_T \odot \boldsymbol{\epsilon}_T$, $\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t \ (t < T), \quad \boldsymbol{x} := \boldsymbol{x}_0.$
Deterministic DPMs	$\boldsymbol{z} := \boldsymbol{x}_T$ ($T = T_g$ if with gradient)	$\boldsymbol{x}_{T-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_T, T)$, $\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_t, t) \ (t < T), \quad \boldsymbol{x} := \boldsymbol{x}_0.$
LDM	$\boldsymbol{z}$ of G_{latent}	$\boldsymbol{z}_0 = G_{\text{latent}}(\boldsymbol{z}), \quad \boldsymbol{x} = D(\boldsymbol{z}_0).$
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	$\boldsymbol{z} := (\boldsymbol{x}_T \oplus \boldsymbol{z}_T \oplus \boldsymbol{\epsilon}_T \oplus \cdots \oplus \boldsymbol{z}_2 \oplus \boldsymbol{\epsilon}_2 \oplus \boldsymbol{z}_1)$	$\boldsymbol{x}_{T-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_T, \boldsymbol{z}_T, T) + \boldsymbol{\sigma}_T \odot \boldsymbol{\epsilon}_T$, $\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_t, \boldsymbol{z}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t \ (1 < t < T)$, $\boldsymbol{x} := \boldsymbol{x}_0 = \boldsymbol{\mu}_T(\boldsymbol{x}_1, \boldsymbol{z}_1, 1).$

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 $\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z}), \boldsymbol{x} = G(\boldsymbol{z})$. The latent prior $p_{\boldsymbol{z}}(\boldsymbol{z})$ is commonly chosen to be the isometric Gaussian distribution. In this section, we show how to unify diffusion models into this family. Overview is shown 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 $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, 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

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

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) generate images with the ODE formulation. Given the mean estimator $\boldsymbol{\mu}_T$, deterministic DPMs generate $\boldsymbol{x} := \boldsymbol{x}_0$ via

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

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

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

Latent diffusion models (LDMs): An LDM (Rombach et al., 2022) first uses a diffusion model G_{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_{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:

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

DDGAN: DDGAN (Xiao et al., 2022) models each reverse time step t as a GAN conditioned on the output of the previous step. We define the latent code $\boldsymbol{z}$ and generation process G of DDGAN as

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

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

Algorithm 1: CycleDiffusion for zero-shot image-to-image translation

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

3.2 DPM-Encoder: A Reconstructable Encoder for Diffusion Models

In this section, we investigate the *encoding* problem, i.e., $\boldsymbol{z} \sim \text{Enc}(\boldsymbol{z}|\boldsymbol{x}, G)$. The encoding problem has been studied for many generative models, and our contribution is **DPM-Encoder**, an encoder for stochastic DPMs. DPM-Encoder is defined as follows. 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 $\boldsymbol{z} \sim \text{DPMEnc}(\boldsymbol{z}|\boldsymbol{x}, G)$ as (see details in Appendices A and B)

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

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

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

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 (Nichol et al., 2022). To formalize "similar images", we provide an upper bound of image distances based on assumptions about the trained DPMs, shown at the end of this subsection. Based on this finding, we propose a simple unpaired image-to-image translation method, 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 $\hat{\boldsymbol{x}} = G_2(\boldsymbol{z})$:

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

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)

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

Inspired by the realism-faithfulness tradeoff in SDEdit (Meng et al., 2022), 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 the following 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

$$B_{t-1} \leq (K_t + 1)B_t + S_t. \tag{9}$$

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). 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 (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

$$\boldsymbol{z}^{\langle k+1 \rangle} = \boldsymbol{z}^{\langle k \rangle} + \frac{\sigma}{2}\nabla_{\boldsymbol{z}}\Big(\log p_{\boldsymbol{z}}(\boldsymbol{z}^{\langle k \rangle}) - E\big(G(\boldsymbol{z}^{\langle k \rangle})|\mathcal{C}\big)\Big) + \sqrt{\sigma}\boldsymbol{\omega}^{\langle k \rangle}, \quad \boldsymbol{\omega}^{\langle k \rangle} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \tag{10}$$

Table 2: 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 ($\eta = 0.1$)	**58.87**	20.3	18.50	**0.557**	**56.45**	19.5	17.82	**0.479**

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 what we call zero-shot image-to-image translation; CycleDiffusion outperforms several image-to-image translation baselines that we re-purposed for this new task. Section 4.3 shows how diffusion models and GANs can be guided 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. 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 256×256 for Cat → Dog and Wild → Dog. For each source image, each method should generate a target image with minimal changes. Since CycleDiffusion sometimes generates noisy outputs, we used T_{sdedit} steps of SDEdit for denoising. When $T = 1000$, we set $T_{\text{sdedit}} = 100$ for Cat → Dog and $T_{\text{sdedit}} = 125$ for Wild → Dog.

Metrics: To evaluate realism, we reported Frechet Inception Distance (FID; Heusel et al., 2017) and Kernel Inception Distance (KID; Bińkowski et al., 2018) between the generated and target images. To evaluate faithfulness, we reported Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM; Wang et al., 2004) 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 (Park et al., 2020), ILVR (Choi et al., 2021), SDEdit (Meng et al., 2022), and EGSDE (Zhao et al., 2022). CUT is based on GAN, and the others use diffusion models.

Pre-trained diffusion models: 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 Choi et al. (2021) for Dog. CycleDiffusion needs diffusion models on both domains, so we trained them on Cat and Wild.

Seen in Table 2 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 from CycleDiffusion, showing that our method can change the domain while preserving local details such as the background, lighting, pose, and overall color the animal.

Figure 2: Unpaired image-to-image translation (Cat → Dog, Wild → Dog) with CycleDiffusion.

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

This section provides 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 3 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 $\hat{\boldsymbol{x}}$. We also demonstrate that CycleDiffusion can be combined with the Cross Attention Control (Hertz et al., 2022) to further preserve the image structure.

Metrics: To evaluate the faithfulness of the generated image to the source image, we reported PSNR and SSIM. To evaluate the authenticity of the generated image to the target text, we reported the CLIP score $\mathcal{S}_{\text{CLIP}}(\hat{\boldsymbol{x}}|\hat{\boldsymbol{t}}) = \cos\left\langle \text{CLIP}_{\text{img}}(\hat{\boldsymbol{x}}), \text{CLIP}_{\text{text}}(\hat{\boldsymbol{t}}) \right\rangle$, where the CLIP embeddings are normalized. We note a trade-off between PSNR/SSIM and $\mathcal{S}_{\text{CLIP}}$: by copying the source image we get high PSNR/SSIM but low $\mathcal{S}_{\text{CLIP}}$, and by ignoring the source image (e.g., by directly generating images conditioned on the target text) we get high $\mathcal{S}_{\text{CLIP}}$ but low PSNR/SSIM. To address this trade-off, we also reported the directional CLIP score (Patashnik et al., 2021) (the CLIP embeddings are normalized):

$$\mathcal{S}_{\text{D-CLIP}}(\hat{\boldsymbol{x}}|\boldsymbol{x}, \boldsymbol{t}, \hat{\boldsymbol{t}}) = \cos\left\langle \text{CLIP}_{\text{img}}(\hat{\boldsymbol{x}}) - \text{CLIP}_{\text{img}}(\boldsymbol{x}), \text{CLIP}_{\text{text}}(\hat{\boldsymbol{t}}) - \text{CLIP}_{\text{text}}(\boldsymbol{t}) \right\rangle. \tag{11}$$

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 C).

Pre-trained text-to-image diffusion models: We used the following text-to-image diffusion models models: (1) `LDM-400M`, a 1.45B-parameter model trained on LAION-400M (Schuhmann et al., 2021), (2) `SD-v1-4`, a 0.98B-parameter Stable Diffusion trained on LAION-5B (Schuhmann et al., 2022).

Results: Table 3 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. For all methods, we find that the pre-trained weights `SD-v1-1` and `SD-v1-4` have better faithfulness than `LDM-400M`. Figure 3 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. See Figure 7 (Appendix E) for qualitative comparisons with the baselines.

CycleDiffusion + Cross Attention Control: Besides fixing the random seed, Hertz et al. (2022) 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 4 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 → elephant, CAC makes the generated elephant to look more like a horse in shape.

Table 3: Zero-shot image editing. 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 C.

	Method	$\mathcal{S}_{\text{CLIP}}\uparrow$	$\mathcal{S}_{\text{D-CLIP}}\uparrow$	PSNR↑	SSIM↑
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-v1-4	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**

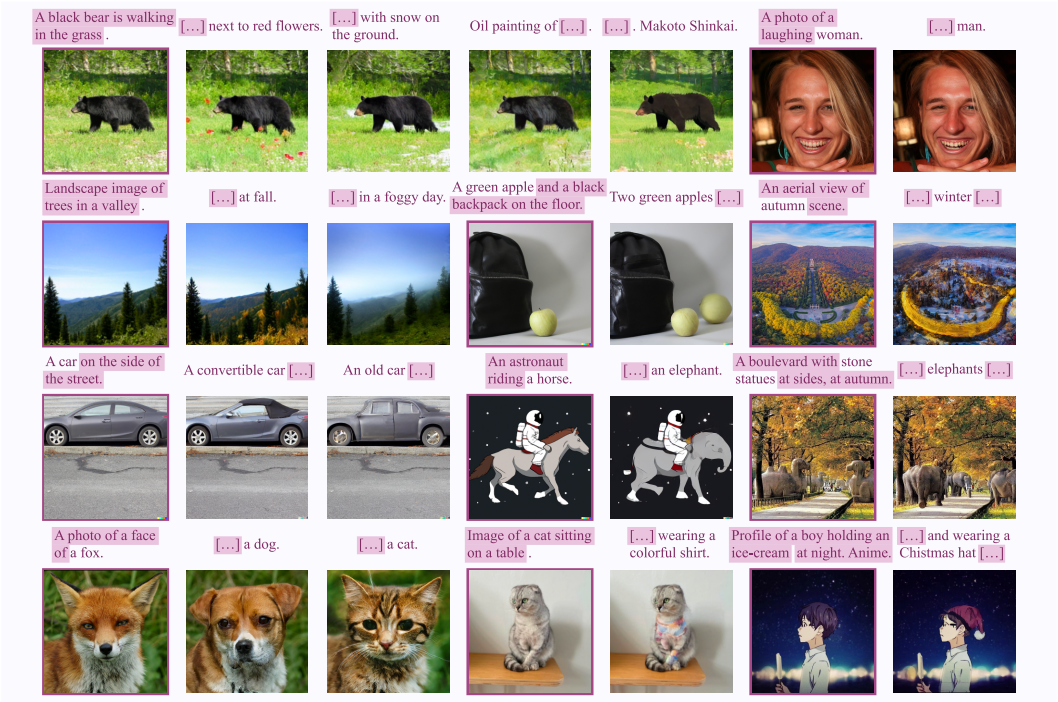


Figure 3: CycleDiffusion for zero-shot image editing. Source images $\boldsymbol{x}$ are displayed with a purple margin; the other images are the generated $\hat{\boldsymbol{x}}$. Within each pair of source and target texts, overlapping text spans are marked in purple in the source text and abbreviated as […] in the target text.

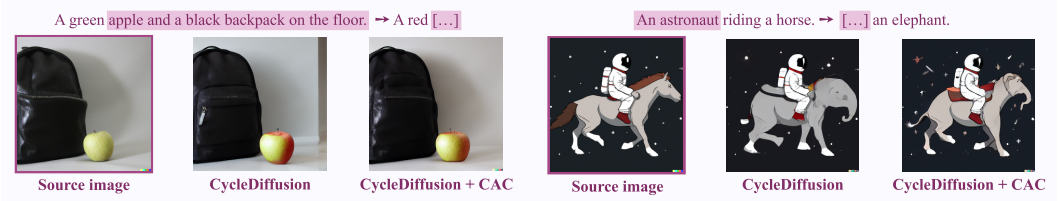
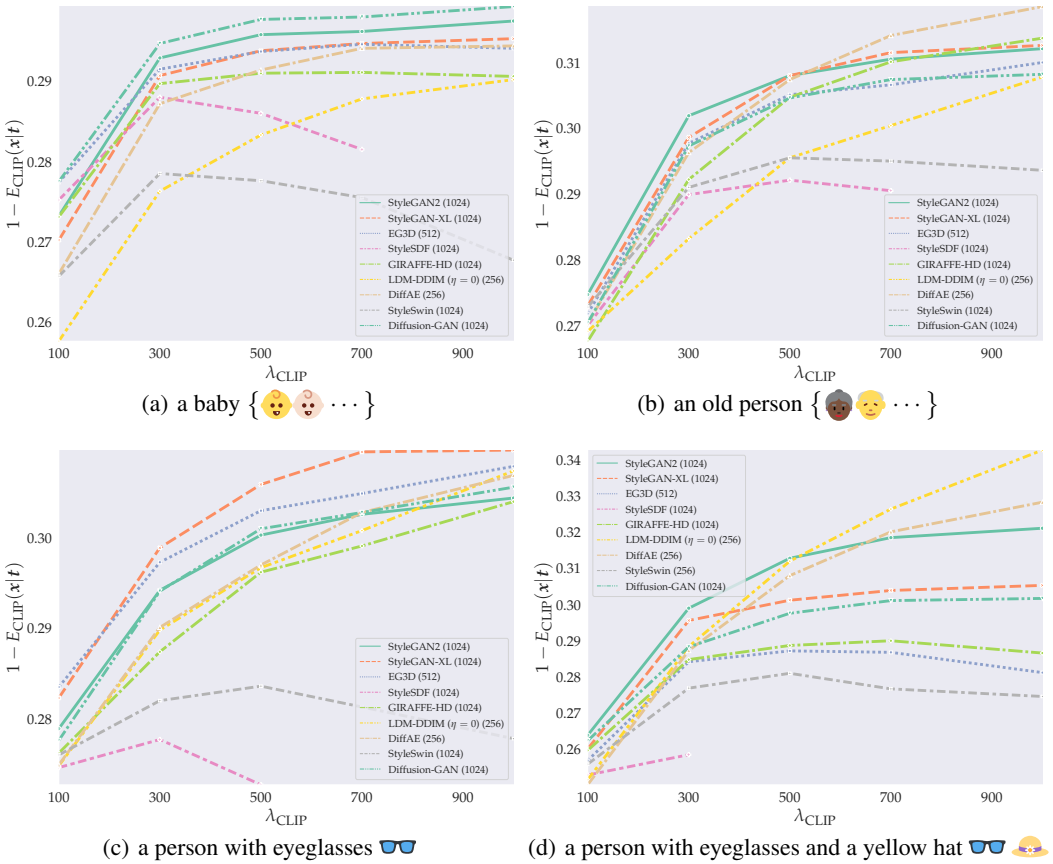


Figure 4: Cross Attention Control (CAC; Hertz et al., 2022) helps CycleDiffusion when the intended *structural* change is small. For instance, when the intended change is color but not shape (left), CAC helps CycleDiffusion preserve the background; when the intended change is horse → elephant, CAC makes the generated elephant look more like a horse in shape.



(a) a baby $\{$ $\cdots\}$ (b) an old person $\{$ $\cdots\}$

(c) a person with eyeglasses (d) a person with eyeglasses and a yellow hat

Figure 5: 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 6 and Appendix G. When the guidance becomes complex, diffusion models surpass GANs.

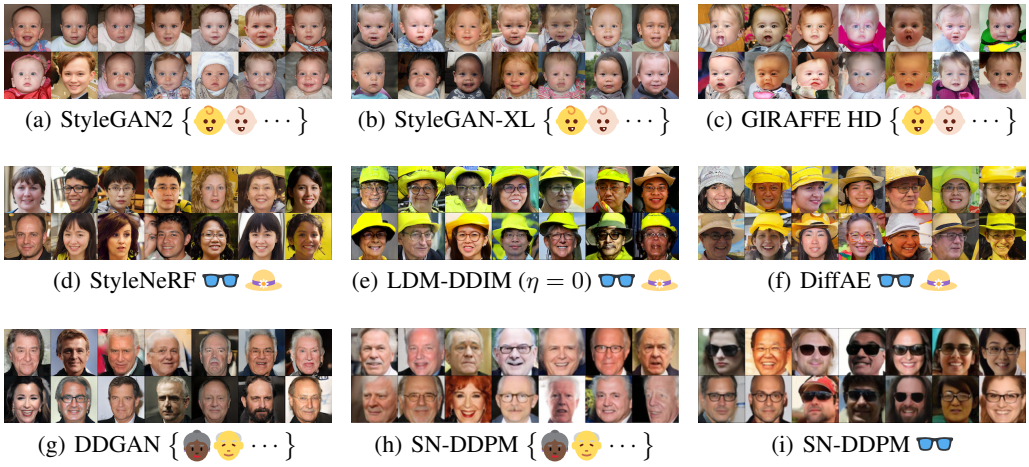
4.3 Unified Plug-and-Play Guidance for Diffusion Models and GANs

Previous methods for conditional sampling from (*aka* guiding) diffusion models require training the guidance model on noisy images (Dhariwal & Nichol, 2021; Liu et al., 2021), which deviates from the idea of plug-and-play guidance by leveraging the simple latent prior of generative models (Nguyen et al., 2017). In contrast, our definition of the Gaussian latent space of different diffusion models allows for unified plug-and-play guidance of diffusion models and GANs. It facilitates principled comparisons over sub-populations and individuals when models are trained on the same dataset.

We used the text $\boldsymbol{t}$ to specify sub-population. For instance, a photo of baby represents the baby sub-population in the domain of human faces. We instantiate the energy in Section 3.4 as $E_{\text{CLIP}}(\boldsymbol{x}|\boldsymbol{t}) = \frac{1}{L}\sum_{l=1}^{L}\left(1 - \cos\left\langle \text{CLIP}_{\text{img}}\left(\text{DiffAug}_l(\boldsymbol{x})\right), \text{CLIP}_{\text{text}}(\boldsymbol{t})\right\rangle\right)$, where DiffAug_l stands for differentiable augmentation (Zhao et al., 2020) that mitigates the adversarial effect, and we sample from the energy-based distribution using Langevin dynamics in Eq. (10) with $n = 200$, $\sigma = 0.05$. We enumerated the guidance strength (i.e., the coefficient λ_c in Section 3.4) $\lambda_{\text{CLIP}} \in \{100, 300, 500, 700, 1000\}$. For evaluation, we reported $(1 - E_{\text{CLIP}}(\boldsymbol{x}|\boldsymbol{t}))$ averaged over 256 samples. This metric quantifies whether the sampled images are consistent with the specified text $\boldsymbol{t}$. Figure 5 plots models with pre-trained weights on FFHQ (Karras et al., 2019) (citations in Table 5, Appendix G). In Figure 6, we visualize samples for SN-DDPM and DDGAN trained on CelebA. We find that diffusion models outperform 2D/3D GANs for complex text, and different models represent the same sub-population differently.

Broad coverage of individuals is an important aspect of the personalized use of generative models. To analyze this coverage, we guide different models to generate images that are close to a reference

$\boldsymbol{x}_r$ in the identity (ID) space modeled by the IR-SE50 face embedding model (Deng et al., 2019), denoted as R. Given an ID reference image $\boldsymbol{x}_r$, we instantiated the energy defined in Section 3.4 as $E_{\text{ID}}(\boldsymbol{x}|\boldsymbol{x}_r) = 1 - \cos\langle R(\boldsymbol{x}), R(\boldsymbol{x}_r)\rangle$ with strength $\lambda_{\text{ID}} = 2500$ (i.e., $\lambda_{\mathcal{C}}$ in Section 3.4). For sampling, we used Langevin dynamics detailed in Eq. (10) with $n = 200$ and $\sigma = 0.05$. To measure ID similarity to the reference image $\boldsymbol{x}_r$, we reported $\cos\langle R(\boldsymbol{x}), R(\boldsymbol{x}_r)\rangle$, averaged over 256 samples. In Table 4, 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. We provide image samples of identity guidance in Figure 9 (Appendix G).



(a) StyleGAN2 $\{$ $\cdots\}$ (b) StyleGAN-XL $\{$ $\cdots\}$ (c) GIRAFFE HD $\{$ $\cdots\}$

(d) StyleNeRF (e) LDM-DDIM ($\eta = 0$) (f) DiffAE

(g) DDGAN $\{$ $\cdots\}$ (h) SN-DDPM $\{$ $\cdots\}$ (i) SN-DDPM

Figure 6: Sampling sub-populations from pre-trained generative models. Notations follow Figure 5.

Table 4: 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 9 (Appendix G).

	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 generative models by reformulating the latent space of diffusion models. While this reformulation is purely definitional, we show that it allows us to use diffusion models in similar ways as CycleGANs (Zhu et al., 2017) and GANs. Our CycleDiffusion achieves impressive performance on unpaired image-to-image translation (with two diffusion models trained on two domains independently) and zero-shot image-to-image translation (with text-to-image diffusion models). Our definition of latent code also allows diffusion models to be guided in the same way as GANs (i.e., plug-and-play, without finetuning on noisy images), and results show that diffusion models have broader coverage of sub-populations and individuals than GANs.

Besides the interesting results, it is worth noting that this paper raised more questions than provided answers. We have provided a formal analysis of the common latent space of stochastic DPMs via the bounded distance between images (Section 3.3), but it still needs further study. Notably, Khrulkov & Oseledets (2022) and Su et al. (2022) studied deterministic DPMs based on optimal transport. Furthermore, *efficient* plug-and-play guidance for stochastic DPMs on high-resolution images with many diffusion steps still remains open. These topics can be further explored in future studies.

REFERENCES

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

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.

Han K. Cao, Cheng Tan, Zhangyang Gao, Guangyong Chen, Pheng-Ann Heng, and Stan Z. Li. A survey on generative diffusion model. *ArXiv*, 2022.

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.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *ArXiv*, 2022.

Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. *UAI*, 2018.

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, 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.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. *ICLR*, 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.

Eliya Nachmani, Robin San-Roman, and Lior Wolf. Non Gaussian denoising diffusion models. *ArXiv*, 2021.

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.

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.

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.

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.

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.

Ling Yang, Zhilong Zhang, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Ming-Hsuan Yang, and Bin Cui. Diffusion models: A comprehensive survey of methods and applications. *ArXiv*, 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.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*, 2017.

A Mathematical Details of Diffusion Models

A.1 Stochastic DPMs

In Eq. (1), we use $\boldsymbol{\mu}_T(\boldsymbol{x}_t, t)$ and $\boldsymbol{\sigma}_t$ as a high-level abstraction to represent each reverse step t (T is the total number of steps) of stochastic DPMs. In Eq. (1), we define the sampling of $\boldsymbol{x}$ as

$$\begin{aligned}
&\boldsymbol{z} := \left(\boldsymbol{x}_T \oplus \boldsymbol{\epsilon}_T \oplus \cdots \oplus \boldsymbol{\epsilon}_2 \oplus \boldsymbol{\epsilon}_1\right) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \\
&\boldsymbol{x}_{T-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_T, T) + \boldsymbol{\sigma}_T \odot \boldsymbol{\epsilon}_T, \\
&\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t, \quad T > t > 0, \\
&\boldsymbol{x} := \boldsymbol{x}_0.
\end{aligned} \tag{12}$$

To be self-contained, here we provide details of $\boldsymbol{\mu}_T(\boldsymbol{x}_t, t)$ and $\boldsymbol{\sigma}_t$ for DDPM (Ho et al., 2020) and DDIM (Song et al., 2021a). Since the notations are not consistent in the two papers, we follow the notation in each paper respectively and use different colors to distinguish different notations. Also note that $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ stands for the neural network learned by DDPM and its variants, which should be distinguished from $\boldsymbol{\epsilon}_t$ used throughout this paper.

DDPM's $\boldsymbol{\mu}_T(\boldsymbol{x}_t, t)$ and $\boldsymbol{\sigma}_t$: We follow the notation in Ho et al. (2020).

$$\boldsymbol{\mu}_T(\boldsymbol{x}_t, t) := \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right), \tag{13}$$

$$\boldsymbol{\sigma}_t := \begin{cases} \sqrt{\beta_t}\boldsymbol{I}, & \text{(option 1)} \\ \sqrt{\dfrac{(1-\bar{\alpha}_{t-1})\beta_t}{1-\bar{\alpha}_t}}\boldsymbol{I}, & \text{(option 2)} \\ \exp\left(\dfrac{\boldsymbol{v}_\theta(\boldsymbol{x}_t, t)}{2}\log\beta_t + \dfrac{\boldsymbol{I} - \boldsymbol{v}_\theta(\boldsymbol{x}_t, t)}{2}\log\dfrac{(1-\bar{\alpha}_{t-1})\beta_t}{1-\bar{\alpha}_t}\right). & \text{(option 3)} \end{cases} \tag{14}$$

DDIM's $\boldsymbol{\mu}_T(\boldsymbol{x}_t, t)$ and $\boldsymbol{\sigma}_t$: We follow the notation in Song et al. (2021a).

$$\boldsymbol{\mu}_T(\boldsymbol{x}_t, t) := \sqrt{\alpha_{t-1}}\left(\frac{\boldsymbol{x}_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t), \tag{15}$$

$$\boldsymbol{\sigma}_t := \sigma_t\boldsymbol{I}, \qquad \text{where } \sigma_t = \eta\sqrt{(1-\alpha_{t-1})/(1-\alpha_t)}\sqrt{1-\alpha_t/\alpha_{t-1}}, \tag{16}$$

where η is a hyper-parameter.

A.2 Deterministic DDIM

Deterministic DDIM's $\boldsymbol{\mu}_T(\boldsymbol{x}_t, t)$: Deterministic DDIM is a special case of DDIM when $\eta = 0$. For details of other deterministic DPMs, please check the original papers.

A.3 Score-based Generative Modeling with SDE

Song et al. (2021b) proposed a unified view of DDPM and score matching with Langevin dynamics (SMLD) as different stochastic differential equations (SDEs). Since the randomness in their sampling algorithms purely come from Gaussian noise, we can incorporate their models and sampling methods into our framework. As a demonstration, we show how to define $\boldsymbol{\mu}_T(\boldsymbol{x}_t, t)$ and $\boldsymbol{\sigma}_t$ for their predictor-only sampling with reverse diffusion samplers. Given a forward SDE:

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, t)\mathrm{d}t + \boldsymbol{\sigma}(t)\odot\mathrm{d}\boldsymbol{w}, \tag{17}$$

the reverse-time SDE is

$$\mathrm{d}\boldsymbol{x} = [\boldsymbol{f}(\boldsymbol{x}, t) - \boldsymbol{\sigma}(t)^2\odot\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x})]\mathrm{d}t + \boldsymbol{\sigma}(t)\odot\mathrm{d}\bar{\boldsymbol{w}}. \tag{18}$$

Suppose the forward SDE is discretized in the following form:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{f}_t(\boldsymbol{x}_t) + \boldsymbol{\sigma}_t\odot\boldsymbol{z}_t, \quad t = 0, \ldots, T-1, \quad \boldsymbol{z}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \tag{19}$$

Reverse diffusion samplers discretize the reserve-time SDE in a similar form:

$$\boldsymbol{x}_{t-1} = \boldsymbol{x}_t - \boldsymbol{f}_t(\boldsymbol{x}_t) + \boldsymbol{\sigma}_t^2\odot\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) + \boldsymbol{\sigma}_t\odot\boldsymbol{\epsilon}_t, \quad t = 1, \ldots, T, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{20}$$

where $\boldsymbol{s}_{\boldsymbol{\theta}}$ is a neural network trained to match the score $\nabla_{\boldsymbol{x}}\log p_t(\boldsymbol{x})$. By comparing Eq. (12) and Eq. (20), we have $\boldsymbol{\mu}_T(\boldsymbol{x}_t, t) := \boldsymbol{x}_t - \boldsymbol{f}_t(\boldsymbol{x}_t) + \boldsymbol{\sigma}_t^2\odot\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$.

A.4 DDGAN

In Eq. (5), we use $\boldsymbol{\mu}_T(\boldsymbol{x}_t, \boldsymbol{z}_t, t)$ and $\boldsymbol{\sigma}_t$ as high-level abstractions to represent each reverse step t (T is the total number of steps) of DDGAN. The generation process is defined as

$$\begin{aligned}
&\boldsymbol{z} := \left(\boldsymbol{x}_T \oplus \boldsymbol{z}_T \oplus \boldsymbol{\epsilon}_T \oplus \cdots \oplus \boldsymbol{z}_2 \oplus \boldsymbol{\epsilon}_2 \oplus \boldsymbol{z}_1\right) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}),\\
&\boldsymbol{x}_{T-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_T, \boldsymbol{z}_T, T) + \boldsymbol{\sigma}_T \odot \boldsymbol{\epsilon}_T,\\
&\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_t, \boldsymbol{z}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t, \quad T > t > 1,\\
&\boldsymbol{x} := \boldsymbol{x}_0 = \boldsymbol{\mu}_T(\boldsymbol{x}_1, \boldsymbol{z}_1, 1).
\end{aligned} \tag{21}$$

To be self-contained, here we provide details of $\boldsymbol{\mu}_T(\boldsymbol{x}_t, \boldsymbol{z}_t, t)$ and $\boldsymbol{\sigma}_t$ of DDGAN.

DDGAN's $\boldsymbol{\mu}_T(\boldsymbol{x}_t, \boldsymbol{z}_t, t)$ and $\boldsymbol{\sigma}_t$: We follow the notation in Xiao et al. (2022) and Ho et al. (2020).

$$\boldsymbol{\mu}_T(\boldsymbol{x}_t, \boldsymbol{z}_t, t) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} G_\theta(\boldsymbol{x}_t, \boldsymbol{z}_t, t) + \frac{\sqrt{\alpha_t}\,(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t, \tag{22}$$

$$\boldsymbol{\sigma}_t := \sqrt{\frac{(1-\bar{\alpha}_{t-1})\beta_t}{1-\bar{\alpha}_t}}\boldsymbol{I}, \tag{23}$$

where $G_\theta(\boldsymbol{x}_t, \boldsymbol{z}_t, t)$ is a conditional GAN learned by DDGAN, which should be distinguished from the deterministic mapping G used throughout this paper.

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

B Mathematical Details of DPM-Encoder

In this section, we provide details of our DPM-Encoder introduced in Section 3.2, which samples $\boldsymbol{z} \sim \mathrm{DPMEnc}(\boldsymbol{z}|\boldsymbol{x}, G)$. For each image $\boldsymbol{x} := \boldsymbol{x}_0$, stochastic DPMs define a posterior distribution over the noisy images $\boldsymbol{x}_{1:T}$, denoted as $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ (Ho et al., 2020; Song et al., 2021a). To be self-contained, we provide details of this posterior distribution for different diffusion models.

DDPM's posterior $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$: We follow the notation in Ho et al. (2020).

$$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) := \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \quad q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) := \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\mathbf{I}\right). \tag{24}$$

DDIM's posterior $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$: We follow the notation in Song et al. (2021a).

$$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) := q(\boldsymbol{x}_T|\boldsymbol{x}_0)\prod_{t=2}^{T} q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0), \tag{25}$$

$$q(\boldsymbol{x}_T|\boldsymbol{x}_0) = \mathcal{N}(\sqrt{\alpha_T}\boldsymbol{x}_0, (1-\alpha_T)\,\boldsymbol{I}), \tag{26}$$

$$\begin{aligned}
q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) &= \mathcal{N}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{x}_0}{\sqrt{1-\alpha_t}}, \sigma_t^2\boldsymbol{I}\right),\\
&\text{where } \sigma_t = \eta\sqrt{(1-\alpha_{t-1})\,/\,(1-\alpha_t)}\sqrt{1-\alpha_t/\alpha_{t-1}}.
\end{aligned} \tag{27}$$

Based on the posterior distribution $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$, DPM-Encoder samples the latent code $\boldsymbol{z}$ by first sampling noisy images $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ from $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ and computing the $\boldsymbol{\epsilon}_t$ according to Eq. (1) and

Eq. (12). Formally, we define the sampling process $\boldsymbol{z} \sim \mathrm{DPMEnc}(\boldsymbol{z}|\boldsymbol{x}, G)$ as

$$\begin{aligned} \boldsymbol{x}_1, \dots, \boldsymbol{x}_{T-1}, \boldsymbol{x}_T \sim q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0), \quad \boldsymbol{\epsilon}_t = \big(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_T(\boldsymbol{x}_t, t)\big)/\boldsymbol{\sigma}_t, \quad t = T, \dots, 1, \\ \boldsymbol{z} := \big(\boldsymbol{x}_T \oplus \boldsymbol{\epsilon}_T \oplus \cdots \oplus \boldsymbol{\epsilon}_2 \oplus \boldsymbol{\epsilon}_1\big). \end{aligned} \tag{28}$$

DPM-Encoder guarantees perfect reconstruction. The proof is straightforward, provided as follows.

Proposition 1. *(Invertibility of DPM-Encoder) For each $\boldsymbol{z} \sim \mathrm{DPMEnc}(\boldsymbol{z}|\boldsymbol{x}, G)$ defined in Eq. (28), we have $\boldsymbol{x} = \bar{\boldsymbol{x}} := G(\boldsymbol{z})$, where $\bar{\boldsymbol{x}} := G(\boldsymbol{z})$ is defined as*

$$\begin{aligned} \bar{\boldsymbol{x}}_{T-1} &= \boldsymbol{\mu}_T(\boldsymbol{x}_T, T) + \boldsymbol{\sigma}_T \odot \boldsymbol{\epsilon}_T, \\ \bar{\boldsymbol{x}}_{t-1} &= \boldsymbol{\mu}_T(\bar{\boldsymbol{x}}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t, \quad T > t > 0, \\ \bar{\boldsymbol{x}} &:= \bar{\boldsymbol{x}}_0. \end{aligned} \tag{29}$$

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

$$\bar{\boldsymbol{x}}_{T-1} = \boldsymbol{\mu}_T(\boldsymbol{x}_T, T) + \boldsymbol{\sigma}_T \odot \boldsymbol{\epsilon}_T \tag{30}$$

$$= \boldsymbol{\mu}_T(\boldsymbol{x}_T, T) + \boldsymbol{\sigma}_T \odot \big(\boldsymbol{x}_{T-1} - \boldsymbol{\mu}_T(\boldsymbol{x}_T, T)\big)/\boldsymbol{\sigma}_T = \boldsymbol{x}_{T-1}. \tag{31}$$

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

$$\bar{\boldsymbol{x}}_{t-1} = \boldsymbol{\mu}_T(\bar{\boldsymbol{x}}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t \tag{32}$$

$$= \boldsymbol{\mu}_T(\boldsymbol{x}_t, t) + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t \tag{33}$$

$$= \boldsymbol{\mu}_T(\boldsymbol{x}_t, t) + \boldsymbol{\sigma}_t \odot \big(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_T(\boldsymbol{x}_t, t)\big)/\boldsymbol{\sigma}_t = \boldsymbol{x}_{t-1}. \tag{34}$$

□

C Experimental Details of Zero-Shot Image-to-Image Translation

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 but are not limited to: (1) image generated by DALL·E 2 (Ramesh et al., 2022), (2) real images from Ruiz et al. (2022), (3) real images from (Hertz et al., 2022), (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.

D Resources

Our experiments used publicly available pre-trained checkpoints (except for the diffusion models trained by us on AFHQ Cat and Wild; see Section 4). Each experiment was run on one NVIDIA

RTX A4000 (16G) / RTX A6000 (48G) / A100 (40G) GPU. Our codes are based on the PyTorch library and are now available at https://github.com/ChenWu98/unified-generative-zoo and https://github.com/ChenWu98/cycle-diffusion.

E Additional Results for Zero-Shot Image-to-Image Translation

Figure 7 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.

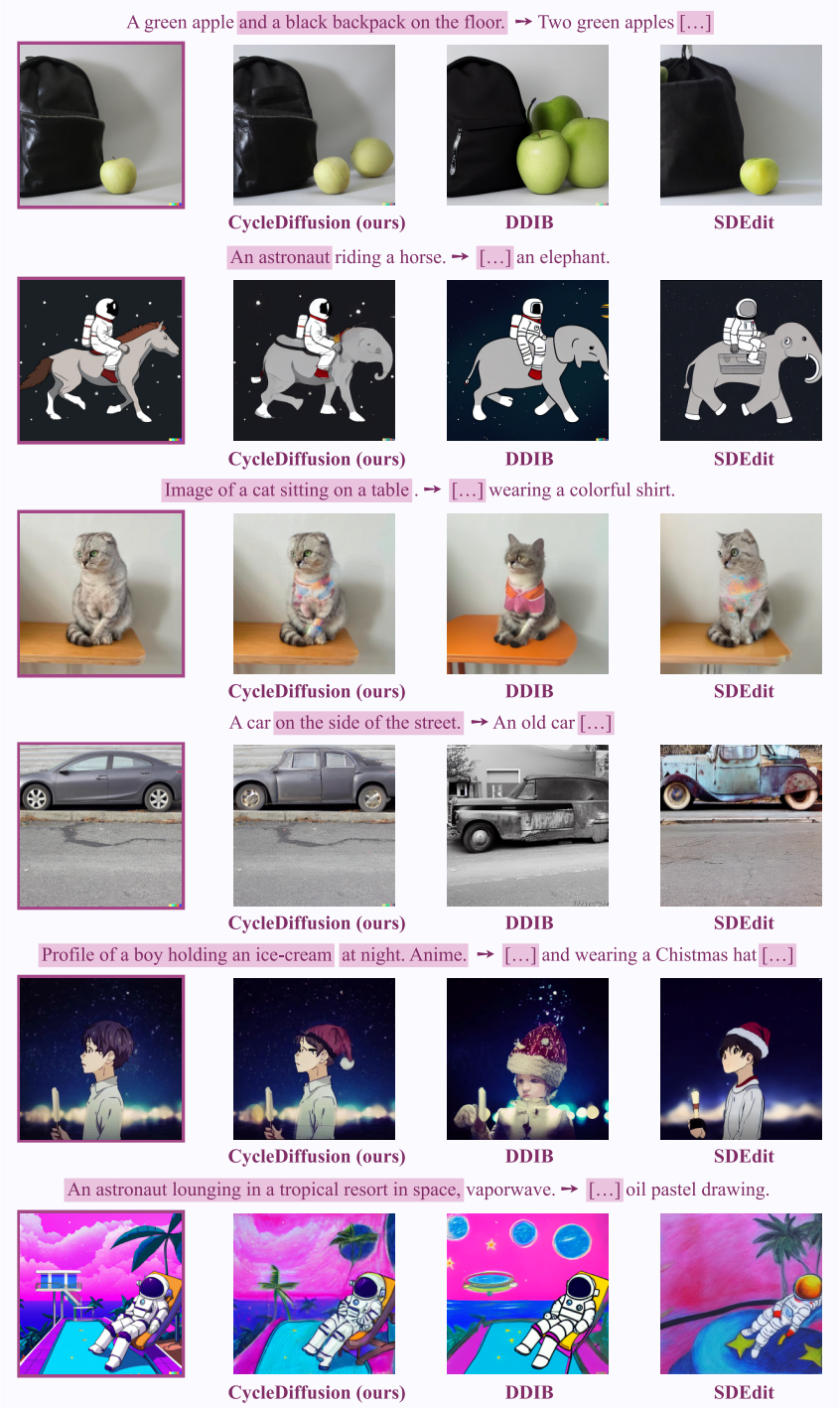


Figure 7: Samples for zero-shot image-to-image translation. Notations follow Figure 3. Compared with DDIB and SDEdit, CycleDiffusion greatly improves the faithfulness to the source image.

F LOCAL EDITING DDIM'S HIGH-DIMENSIONAL LATENT CODE

Local editing of low-dimensional latent code has been shown to be useful for semantic-level image manipulation (Shen et al., 2022). However, it is unclear whether we can perform semantic-level image manipulation via local editing in the high-dimensional latent space diffusion models. Note that this is different from mask-then-inpaint (Ramesh et al., 2022), edit-with-scribbles (Meng et al., 2022), or domain adaptation (Kim et al., 2022b)). Notably, it does not need the classifier to be adapted to noisy images as done by the classifier guidance (Dhariwal & Nichol, 2021; Liu et al., 2021).

Given an image $\boldsymbol{x}_{ori}$, we encode it as $\boldsymbol{z}_{ori}$, edit it as $\boldsymbol{z}_{edit} = \boldsymbol{z}_{ori} + \boldsymbol{n}$, and compute the edited image $\boldsymbol{x}_{edit} = G(\boldsymbol{z}_{edit})$. To learn the vector $\boldsymbol{n}$ for a target class a, we optimize

$$\arg\min_{\|\boldsymbol{n}\|_2=r} \mathbb{E}_{\boldsymbol{z}_{ori}\sim p_{\boldsymbol{z}}(\boldsymbol{z}_{ori}),\ \boldsymbol{z}_{edit}=\boldsymbol{z}_{ori}+\boldsymbol{n}} \Big[-\lambda_{cls} \log P\big(a|G(\boldsymbol{z}_{edit})\big) - \cos\big\langle R(G(\boldsymbol{z}_{edit})), R(G(\boldsymbol{z}_{ori}))\big\rangle \Big], \quad (35)$$

where $P(\cdot|\boldsymbol{x})$ is a classifier trained on CelebA (Liu et al., 2015), and R is the IR-SE50 face embedding model (Deng et al., 2019) to preserve the identity. Empirically, we find that LDM-DDIM ($\eta = 0$) works the best for local editing, as shown in Figure 8.

Figure 8: Image manipulation by local editing of diffusion models' latent code. The diffusion model used here is the deterministic LDM-DDIM ($\eta = 0$).

Table 5: State-of-the-art generative models used in this paper. Notations: *struc.*, –, and $\oplus$ stand for *structure*, *no "latent codes"*,[2] and *progressive generation*, respectively.

	Model name	Latent prior	Objective	Architecture	Latent struc.	Resolution
Diffusion	DDPM (Ho et al., 2020) *etc.*	–	ELBO	{CNN, ViT}	–	256
	DDIM ($\eta = 0$) (Song et al., 2021a)	Gaussian	ELBO		spatial	256
	SN-DDPM (Bao et al., 2022)	–	ELBO		–	64
	ScoreSDE (Song et al., 2021b)	–	ELBO / SM		–	256 / 1024
	LDM (Rombach et al., 2022) *etc.*	diffusion	ELBO		spatial	256
	DiffAE (Preechakul et al., 2022)	diffusion	ELBO		hybrid	256
	DDGAN (Xiao et al., 2022)	–	hybrid		–	256
2D GAN	StyleGAN2 (Karras et al., 2020)	Gaussian	GAN	CNN	vector	1024
	StyleGAN-XL (Sauer et al., 2022)		GAN	CNN		256 – 1024
	StyleSwin (Zhang et al., 2022a)		GAN	ViT		256 / 1024
	BigGAN (Brock et al., 2019)		GAN	CNN		256
	Diffusion-GAN (Wang et al., 2022)		hybrid	CNN		1024
3D GAN	StyleNeRF (Gu et al., 2022)	Gaussian	GAN	NeRF $\oplus$ CNN	vector	256 – 1024
	GIRAFFE-HD (Xue et al., 2022)			NeRF $\oplus$ CNN		1024
	StyleSDF (Or-El et al., 2022)			SDF $\oplus$ CNN		512 / 1024
	EG3D (Chan et al., 2022)			TriPl $\oplus$ CNN		512
VAE	NVAE (Vahdat & Kautz, 2020)	Gaussian	ELBO	CNN	spatial	256

G ADDITIONAL RESULTS FOR PLUG-AND-PLAY GUIDANCE

Seen in Table 5 is a summary of generative models unified as deterministic mappings in this paper. Different models have different training objectives, model architectures, and structures of "latent code"[2]. Most of the listed models are included in our experiments. Table 6 and Table 7 provide a more detailed version of the results (for some generative models) seen in Figure 5. Specifically, we investigated different configurations of various diffusion models and GANs. In Figure 9, we provide several image samples for ID-controlled sampling from pre-trained generative models. Consistent with Table 4, diffusion models have better coverage of individuals than 2D/3D GANs.

Table 6: CLIP experiments of models that have different configurations. Numbers under each model stand for the image resolution; *trunc.* $\phi = 0.7$ stands for the truncation trick (Karras et al., 2019) with truncation coefficient $\phi = 0.7$. The reported metric is the CLIP score (larger is better), the same as Figure 5. $^\heartsuit$ and $^\spadesuit$ stand for the configuration plotted in Figure 5.

Text t (Figure 5)	$\{$ 👶 $\cdots\}^\heartsuit$					$\{$ 👨 $\cdots\}^\spadesuit$				
Control strength λ_{CLIP}	100	300	500	700	1000	100	300	500	700	1000
LDM-DDIM ($\eta = 0$)										
256 ($T_g = 10$)$^{\heartsuit\spadesuit}$	0.258	0.276	0.283	0.288	0.290	0.269	0.283	0.296	0.300	0.308
256 ($T_g = 5$)	0.257	0.280	0.283	0.285	0.287	0.269	0.283	0.292	0.298	0.304
DiffAE										
256 ($T_g = 10$)$^{\heartsuit\spadesuit}$	0.266	0.287	0.291	0.294	0.294	0.270	0.296	0.307	0.314	0.319
128 ($T_g = 3$)	0.259	0.284	0.289	0.290	0.292	0.256	0.271	0.286	0.293	0.298
128 ($T_g = 3$, z_T only)	0.256	0.289	0.289	0.290	0.295	0.256	0.270	0.285	0.293	0.297
StyleGAN2										
1024$^{\heartsuit\spadesuit}$	0.273	0.293	0.296	0.296	0.298	0.275	0.302	0.308	0.311	0.312
1024 (*trunc.* $\phi = 0.7$)	0.267	0.287	0.291	0.293	0.293	0.267	0.291	0.299	0.301	0.303
StyleGAN-XL										
1024$^{\heartsuit\spadesuit}$	0.270	0.291	0.294	0.295	0.295	0.273	0.299	0.308	0.312	0.313
1024 (*trunc.* $\phi = 0.7$)	0.263	0.283	0.287	0.289	0.290	0.265	0.284	0.292	0.295	0.297
512 (*trunc.* $\phi = 0.7$)	0.263	0.282	0.286	0.288	0.289	0.263	0.284	0.293	0.296	0.300
256 (*trunc.* $\phi = 0.7$)	0.262	0.281	0.284	0.287	0.289	0.259	0.281	0.291	0.295	0.299
StyleSwin										
1024$^{\heartsuit\spadesuit}$	0.266	0.279	0.278	0.276	0.268	0.273	0.291	0.296	0.295	0.294
256	0.262	0.282	0.283	0.283	0.281	0.267	0.285	0.290	0.293	0.293
1024 (*trunc.* $\phi = 0.7$)	0.265	0.284	0.287	0.288	0.288	0.264	0.279	0.292	0.297	0.300
256 (*trunc.* $\phi = 0.7$)	0.259	0.278	0.281	0.275	0.273	0.261	0.276	0.281	0.284	0.281
Diffusion-GAN										
1024$^{\heartsuit\spadesuit}$	0.278	0.295	0.298	0.298	0.299	0.270	0.297	0.305	0.307	0.308
1024 (*trunc.* $\phi = 0.7$)	0.273	0.294	0.294	0.300	0.301	0.262	0.286	0.300	0.305	0.309
StyleNeRF										
1024	0.246	0.271	0.283	0.288	0.291	0.264	0.291	0.303	0.307	0.311
256	0.234	0.240	0.247	0.252	0.259	0.260	0.275	0.291	0.298	0.303
1024 (*trunc.* $\phi = 0.7$)	0.243	0.266	0.277	0.283	0.287	0.260	0.280	0.291	0.296	0.300
256 (*trunc.* $\phi = 0.7$)	0.229	0.235	0.239	0.243	0.249	0.255	0.267	0.282	0.290	0.295
StyleSDF										
1024$^{\heartsuit\spadesuit}$	0.275	0.288	0.286	0.282	–	0.270	0.290	0.292	0.291	–
1024 (*trunc.* $\phi = 0.7$)	0.267	0.283	0.284	0.279	–	0.261	0.270	0.273	0.273	–
EG3D										
512$^{\heartsuit\spadesuit}$	0.277	0.292	0.294	0.295	0.294	0.272	0.298	0.305	0.307	0.310
512 (*trunc.* $\phi = 0.7$)	0.277	0.270	0.276	0.280	0.283	0.265	0.285	0.293	0.296	0.298

Figure 9: Image samples for the face ID experiment in Table 4.

Table 7: CLIP experiments of models that have different configurations. Numbers under each model stand for the image resolution; *trunc.* $\phi = 0.7$ stands for the truncation trick (Karras et al., 2019) with truncation coefficient $\phi = 0.7$. The reported metric is the CLIP score (larger is better), the same as Figure 5. $^\heartsuit$ and $^\spadesuit$ stand for the configuration plotted in Figure 5.

Text t (Figure 5)	👓 $^\heartsuit$					👓 🧢 $^\spadesuit$				
Control strength λ_{CLIP}	100	300	500	700	1000	100	300	500	700	1000
LDM-DDIM ($\eta = 0$)										
256 ($T_g = 10$)$^{\heartsuit\spadesuit}$	0.275	0.290	0.297	0.301	0.307	0.252	0.288	0.312	0.326	0.343
256 ($T_g = 5$)	0.273	0.289	0.300	0.305	0.310	0.250	0.288	0.315	0.329	0.343
DiffAE										
256 ($T_g = 10$)$^{\heartsuit\spadesuit}$	0.275	0.290	0.297	0.303	0.307	0.250	0.287	0.308	0.320	0.328
128 ($T_g = 3$)	0.265	0.281	0.288	0.293	0.298	0.240	0.273	0.300	0.314	0.326
128 ($T_g = 3$, z_T only)	0.263	0.280	0.288	0.291	0.296	0.240	0.275	0.300	0.313	0.324
StyleGAN2										
1024$^{\heartsuit\spadesuit}$	0.279	0.294	0.300	0.303	0.304	0.264	0.299	0.313	0.319	0.321
1024 (*trunc.* $\phi = 0.7$)	0.278	0.291	0.297	0.300	0.303	0.255	0.286	0.303	0.311	0.316
StyleGAN-XL										
1024$^{\heartsuit\spadesuit}$	0.282	0.299	0.306	0.310	0.310	0.260	0.296	0.301	0.304	0.305
1024 (*trunc.* $\phi = 0.7$)	0.281	0.297	0.303	0.305	0.309	0.253	0.279	0.287	0.288	0.291
512 (*trunc.* $\phi = 0.7$)	0.280	0.296	0.301	0.305	0.307	0.250	0.282	0.296	0.300	0.303
256 (*trunc.* $\phi = 0.7$)	0.275	0.290	0.297	0.299	0.303	0.251	0.283	0.295	0.300	0.304
StyleSwin										
1024$^\heartsuit$	0.276	0.282	0.284	0.281	0.278	0.251	0.263	0.258	0.255	0.247
256$^\spadesuit$	0.273	0.281	0.285	0.284	0.281	0.256	0.277	0.281	0.277	0.275
1024 (*trunc.* $\phi = 0.7$)	0.276	0.284	0.286	0.290	0.288	0.243	0.263	0.274	0.277	0.275
256 (*trunc.* $\phi = 0.7$)	0.272	0.280	0.281	0.282	0.281	0.248	0.267	0.275	0.274	0.269
Diffusion-GAN										
1024$^{\heartsuit\spadesuit}$	0.278	0.294	0.301	0.303	0.306	0.262	0.288	0.298	0.301	0.302
1024 (*trunc.* $\phi = 0.7$)	0.277	0.291	0.300	0.291	0.308	0.249	0.279	0.289	0.296	0.301
StyleNeRF										
1024	0.268	0.277	0.282	0.285	0.287	0.238	0.252	0.262	0.272	0.281
256	0.264	0.272	0.277	0.280	0.283	0.235	0.244	0.252	0.256	0.263
1024 (*trunc.* $\phi = 0.7$)	0.268	0.276	0.281	0.284	0.286	0.233	0.244	0.253	0.261	0.270
256 (*trunc.* $\phi = 0.7$)	0.264	0.271	0.277	0.280	0.282	0.232	0.238	0.246	0.251	0.255
StyleSDF										
1024$^{\heartsuit\spadesuit}$	0.275	0.278	0.273	–	–	0.253	0.259	–	–	–
1024 (*trunc.* $\phi = 0.7$)	0.273	0.279	0.275	–	–	0.242	0.252	0.248	–	–
EG3D										
512$^{\heartsuit\spadesuit}$	0.284	0.297	0.303	0.305	0.308	0.257	0.284	0.287	0.287	0.281
512 (*trunc.* $\phi = 0.7$)	0.282	0.295	0.300	0.301	0.305	0.246	0.268	0.276	0.278	0.276

H Societal Impact

In general, improved generative modeling makes it easier to generate fake media (e.g., DeepFakes; Westerlund, 2019; Vaccari & Chadwick, 2020) and privacy leaks (e.g., identity-conditioned human face synthesis, information leaks from large-scale pre-training data of text-to-image diffusion models). Additionally, in the particular case of this paper, one could encounter biases image editing as a result of applying CycleDiffusion to text-to-image diffusion models that reflect the natural biases in large text-image pre-training data. On the other hand, improved generative modeling can bring benefits to synthesis of humans and new ways of human communication in AR/VR. Moreover, we point out that there exist many current research works and tools that can efficiently detect fake media or can manage privacy leaks during pre-training. We encourage researchers and practitioners to consider these risks and remedies when using the methods developed in this paper.