Appendices for Boundary Guided Mixing Trajectory for Semantic Control with Diffusion Models

Anonymous Author(s)
Affiliation
Address
email

We present the detailed related work about the denoising diffusion models including the marginal discussion in Appendix A. The high-dimensional space properties and lemmas are introduced in Appendix B, and we explicitly describe how those established theoretical theorems are used and connected to our analysis in the main paper Sec. 3. In Appendix C, we provide the theoretical foundations of the Markov mixing study, which inspires us to formulate the mixing step problem for DDMs. More details about the mixing step problem, formulation, proof and discussion are included in Appendix D. Appendix E includes details about our semantic boundary search method and further discussions in terms of two latent space levels (*e.g.*, generic ϵ-space and h-space from the U-Neu bottleneck [21]). We show the algorithm of our proposed boundary-guided mixing trajectory method in Appendix F. More **randomly selected and non cherry-picked** experimental results, details about user study, and some failure cases analysis are shown in Appendix G. Final discussions about the limitations, time and resource cost, as well as an extended broader impact are included in Appendix H.

A Detailed Related Work

A.1 Denoising Diffusion Models

While we have briefly introduced the preliminaries on DDPMs [15] and DDIMs [32] in the main paper, we re-organize and present more details here. We note that the relevant background is mainly from the original papers, we only include the relevant background information to better illustrate our ideas in this work.

The key idea for generative tasks is to approximate a data distribution $q(x_0)$ with a model learned distribution $p_\theta(x_0)$ that can be easily sampled from. The original Denoising Diffusion Probabilistic Models (DDPMs) [31] propose to use latent variable models to fulfill the goal with the following specific form:

$$p_\theta := \int p_\theta(x_{0:T}) dx_{1:T}, \tag{1}$$

where $x_1, ..., x_T$ are variables modeled by the latent states of a Markov chain, which have the same dimensionality as the actual data $x_0 \sim q(x_0)$. Specifically, we have:

$$p_\theta(x_{0:T}) := p_\theta(x_T) \prod_{t=1}^{T} p_\theta^{(t)}(x_{t-1}|x_t). \tag{2}$$

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

Figure 1: **Non-Cherry-Picked** randomly selected results for *add smiling* and *remove smiling* editing operations from our proposed *BoundaryDiffusion*, Asyrp [21], and DiffusionCLIP [20].

The training objective is the variational lower bound on negative log likelihood:

$$\begin{aligned} L &:= \mathbb{E}[-\log p_\theta(x_0)] \\ &\leq \mathbb{E}[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}] \\ &= \mathbb{E}_q[-\log p(x_T) - \sum_{t\geq 1} \log \frac{p_{\theta(x_{t-1}|x_t)}}{q(x_t|x_{t-1})}]. \end{aligned} \tag{3}$$

The above formulation indicates that the DDPMs can be learned with a pre-defined inference procedure $q(x_{1:T}|x_0)$. In the case of [15], the authors propose to model the Markov chain with Gaussian transitions parameterized by a decreasing sequence $\alpha_{1:T} \in (0, 1]^T$ as follows:

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}), \tag{4}$$

where $q(x_t|x_{t-1}) := \mathcal{N}(\sqrt{\frac{\alpha_t}{\alpha_{t-1}}}x_{t-1}, (1 - \frac{\alpha_t}{\alpha_{t-1}})\mathbf{I})$.

We often refer to the above-mentioned processes from x_0 to x_T and from x_T to x_0 as *forward process* and *reverse process* (or *generative process*), respectively. Intuitively, the forward process adds noise to data x_0, while the reverse process denoises a noisy latent variable $x_{1:T}$. The reverse denoising is stochastic based on this formulation.

A.2 Marginal Discussion for Deterministic Inversion

Motivated to reduce the iteration numbers from the original DDPMs [31, 15], Denoising Diffusion Implicit Models (DDIMs) [32] propose to generalize the inference process (*i.e.*, forward process) from a Markov chain to a Non-Markov one. The theoretical support for the proposed generalization lies within the fact the learning objective of DDPMs only depends on the conditional (on x_0) marginals $q(x_t|x_0)$, instead of the conditional (on x_0) joint $q(x_{1:T}|x_0)$.

Based on the previous fact, DDIMs consider a family of inference distribution $\mathcal{Q}$, indexed by a real vector $\sigma \in \mathbb{R}^T_{\geq 0}$:

$$q_\sigma(x_{1:T}|x_0) := q_\sigma(x_T|x_0) \prod_{t=2}^{T} q_\sigma(x_{t-1}|x_t, x_0), \tag{5}$$

where $q_\sigma(x_T|x_0) = \mathcal{N}(\sqrt{\alpha_T}x_0, (1 - \alpha_T)\mathbf{I})$. Specifically, the $q_\sigma(x_{t-1}|x_t, x_0)$ is carefully designed in a way that the mean function satisfies the above Gaussian kernel as:

$$q_\sigma(x_{t-1}|x_t, x_0) = \mathcal{N}(\sqrt{\alpha_{t-1}}x_0 + \sqrt{1 - \alpha_{t-1} - \sigma_t^2}\frac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1 - \alpha_t}}, \sigma_t^2\mathbf{I}). \tag{6}$$

Using the Bayes' rule, Eq. 6 can be further rewritten as:

$$q_\sigma(x_t|x_{t-1}, x_0) = \frac{q_\sigma(x_{t-1}|x_t, x_0)q_\sigma(x_t|x_0)}{q_\sigma(x_{t-1}|x_0)}. \tag{7}$$

The above Eq. 6 and Eq. 7 show that the Non-Markov process q_σ considered in DDIMs is marginal and also Gaussian (but not a standard one).

After having specified the forward process, DDIMs propose a different variant of the sampling process where the model is expected to first predict the corresponding noiseless x_0 given a noisy observation x_t, and use the prediction to obtain x_{t-1} through Eq. 7. Specifically, the iteration can be written as follows:

$$x_{t-1} = \sqrt{\alpha - 1}(\frac{x_t - \sqrt{1 - \alpha_t}\varepsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}}) + \sqrt{1 - \alpha_{t-1} - \sigma_t^2}\varepsilon_\theta^{(t)}(x_t) + \sigma_t\varepsilon_t, \tag{8}$$

where $\varepsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. By choosing the $\sigma_t = 0$ for all steps t, the random noise induced by the last term from Eq. 8 is removed, and therefore changing the stochastic process from the original DDPMs formulation to a deterministic one.

By connecting the Eq. 8 to the Euler integration for solving ordinary differential equations (ODEs), it can be further rewritten as:

$$\frac{x_{t-1}}{\sqrt{\alpha_{t-1}}} = \frac{x_t}{\sqrt{\alpha_t}} + \left(\sqrt{\frac{1-\alpha_{t-1}}{\alpha_{t-1}}} - \sqrt{\frac{1-\alpha_t}{\alpha_t}}) \right) \epsilon^t(x_t), \tag{9}$$

which is the Euler solution for the following:

$$d\bar{x}(t) = \varepsilon_\theta^{(t)}(\frac{\bar{x}(t)}{\sqrt{\sigma^2+1}})d\sigma(t). \tag{10}$$

Therefore, when we adopt the deterministic inversion method to convert x_0 to x_T, we preserve the marginal property of the considered family of inference distribution $\mathcal{Q}$. Intuitively, extending this distribution family to the d-dimensional space, it ensembles a group of Gaussian distributions parameterized by the defined α sequence. Given the denoising process can be considered as a trajectory, the deterministic inversion follows and stays at the border of the space ensemble.

A.3 Other Related Works

Other studies related to this work include the areas of GANs inversion, image editing and manipulations.

GAN Inversion. GAN inversion problem [35] is proposed to tackle the lack of inference capability in GANs [12]. As another powerful model other than DMs for data generation, many GAN models [17, 9, 18, 19] have been proposed for high-quality image synthesis. With high-level objectives to invert a given image and to apply it in downstream tasks like image editing, the two problems are often studied separately. Specifically, due to the intractability of GAN generation, many works have been focused solely on the first objective to invert an image to the latent space and to reconstruct from the latent encoding, which corresponds to the initial and primary goal of GAN inversion. There are three main technical directions for the inversion and reconstruction problem, which consists of learning an additional deterministic encoder [38, 28, 34, 3], directly solving the optimization problem [1, 2, 16, 11], or a hybrid way that combines the above two techniques [37, 7, 6].

Different from existing GAN inversion works, we leverage the better tractability of DMs and use the deterministic property from the denoising diffusion implicit models (DDIMs) [32] to achieve the inversion and reconstruction when studying the diffusion direction.

Image Manipulation and CLIP Guidance. Image manipulation based on generative models mainly covers two categories. While one branch of existing works often requires retraining of a generative model (*e.g.*, GANs [12]) [26, 33, 22, 5], others are studied as a downstream task application for GAN inversion works [37, 30, 18, 1]. For image manipulation using the GAN inversion technique, a prerequisite for effective editing is a disentangled understanding of latent spaces from pre-trained GAN models. The analysis on the latent space addresses several different separate latent spaces such as the $\mathcal{Z}$ space for generic GANs [12] and the $\mathcal{W}$ space from StyleGAN [18]. The current SOTA methods for diffusion-based editing like DiffusionCLIP [20] and Aysrp [21] all adopt the CLIP guidance as part of their loss function during the learning process.

In this work, we adopt a similar semantic disentanglement idea as the tool to interpret and understand the latent space along the chain. At the same time, we are able to leverage our analysis and a better understanding of the latent space to achieve real-face image editing.

B High Dimensional Space

In this section, we provide the necessary theoretical foundations for understanding the geometric and probabilistic properties of high-dimensional spaces. The majority of the properties and lemmas we describe here are established theorems from high-dimensional space studies in mathematics and statistics from [8]. We omit the detailed proofs for the following properties and lemmas, and kindly ask readers to refer to the original book if interested.

Property B.1. *For a unit-radius sphere in high dimensions, as the dimension d increases, the volume of the sphere goes to 0, and the maximum possible distance between two points stays at 2.*

Lemma B.2. *The surface area $A(d)$ and the volume $V(d)$ of a unit-radius sphere in d-dimensions are given by:*

$$A(d) = \frac{2\pi^{d/2}}{\Gamma(d/2)}, V(d) = \frac{\pi^{d/2}}{\frac{d}{2}\Gamma(d/2)}, \tag{11}$$

where $\Gamma(x)$ is a generalization of the factorial function for noninteger values of x.

The above Property B.1 and Lemma B.2 are generic geometric properties for high-dimensional spheres, but also applicable to high-dimensional Gaussian in which we are interested in the context of DDMs. To draw the connections with our context for studying the latent spaces of DMMs, with higher dimensionality, the latent Gaussian spaces of pre-trained DDMs become more difficult to operate due to decreased volume and mass concentration, as empirically suggested in [20, 21].

Property B.3. *The volume of a high-dimensional sphere is essentially all contained in a thin slice at the equator and is simultaneously contained in a narrow annulus at the surface, with essentially no interior volume. Similarly, the surface area is essentially all at the equator.*

The Property B.3 implies the connection with the standard Gaussian in ϵ_T from direct sampling. In Fig.2 (b) of the main paper where we illustrate the geometric and probabilistic properties of samples in the ϵ_T space, the inverted ones locate in the inner border area of the narrow annulus, which is also empirically verified in our Tab. 1 in the main paper. As those inverted latent encodings have a smaller radius than the expected standard Gaussian case.

Lemma B.4. *For any $c > 0$, the fraction of the volume of the hemisphere above the plane $x_1 = \frac{c}{\sqrt{d-1}}$ is less than $\frac{2}{c}e^{-\frac{c^2}{2}}$.*

The above Lemma B.4 explains the volume range we show in Fig.2 (b) of the main paper in the left side of the Gaussian sphere to show the concentration mass, which is in the order of $O(\frac{r}{\sqrt{d}})$.

Lemma B.5. *The maximum likelihood spherical Gaussian for a set of samples is the one over center equal to the sample mean and standard deviation equal to the standard deviation of the sample.*

The above Lemma B.5 provides the theoretical justifications for using the mean of squared distance to estimate the radius of Gaussian high-dimensional space.

C Markov Mixing

The mixing time defines a parameter that measures the time required by a Markov chain for the distance to stationary to be small [23]. The study of Markov mixing time aims to quantify the speed of convergence for Markov chains, and requires some other necessary preliminary knowledge on the *total variance distance* and the Convergence Theorem, which we will briefly describe below.

Firstly, to quantify the convergence characteristic of Markov chains, an appropriate distance measure metric is a prerequisite. In the literature, the *total variation distance* is the metric used to define the distance.

Definition C.1. The total variation distance between two probability distributions μ and ν on $\mathcal{X}$ is defined by:

$$||\mu - \nu||_{TV} = \max_{A \subseteq \mathcal{X}} |\mu(A) - \nu(A)|. \tag{12}$$

In the above definition, A is a probabilistic event, indicating that the distance between μ and ν is the maximum difference between probabilities assigned to a single event by the two distributions. This initial definition is not very practical to estimate the actual distance, which further induces the following propositions.

Proposition C.2. *Let μ and ν to be two probability distributions on $\mathcal{X}$. Then*

$$||\mu - \nu||_{TV} = \frac{1}{2} \sum_{x \in \mathcal{X}} |\mu(x) - \nu(x)|. \tag{13}$$

Proof. Let $B = \{x : \mu(x) \geq \nu(x)\}$ and let $A \subset \mathcal{X}$ be any event. Then we have

$$\mu(A) - \nu(A) \leq \mu(A \cap B) \leq \mu(B) - \nu(B). \tag{14}$$

The first inequality holds since any $x \in A \cap B^c$ satisfies $\mu(x) - \nu(x) < 0$, and thus the difference in probability cannot decrease when such elements of B are eliminated. For the second inequality, we note that including more elements of B can not decrease the difference in probability.

By the same reasoning, we have:

$$\nu(A) - \mu(A) \leq \nu(B^c) - \mu(B^c). \tag{15}$$

The upper bounds on the right sides of Equation (14) and (15) are the same. Furthermore, by taking $A = B$ or $A = B^c$, then $|\mu(A) - \nu(A)|$ is equal to the upper bound. Therefore, we arrive at:

$$||\mu - \nu||_{TV} = \frac{1}{2}(\mu(B) - \nu(B) + \nu(B^c) - \mu(B^c)) = \frac{1}{2}\sum_{x \in \mathcal{X}} |\mu(x) - \nu(x)|. \tag{16}$$

□

The above proposition reduces total variance distance to a simple sum over the state space, which is an important theoretical support to formulate our mixing step problem and empirical search method. The proof process also reveals the following remark.

Remark C.3. $||\mu - \nu||_{TV} = \sum_{x \in \mathcal{X}, \mu(x) \geq \nu(x)} [\mu(x) - \nu(x)]$.

We then proceed to introduce the convergence theorem, which claims that aperiodic Markov chains converge to their stationary distributions at a key step, which is the direct theoretical foundation for us to introduce the mixing step problem for DDMs.

Theorem C.4. ***Convergence Theorem*** *Suppose that P is irreducible and aperiodic, with stationary distribution π. Then there exit constants $\alpha \in (0, 1)$ and $C > 0$ such that:*

$$max_{x \in \mathcal{X}} ||P^t(x, .) - \pi||_{TV} \leq C\alpha^t. \tag{17}$$

There exist multiple mathematical versions for the proof of the convergence theorem, which we omit in this appendix. Note that the assumptions for P to be irreducible and aperiodic are essential. We recall here the definition of an *irreducible* chain P.

Definition C.5. A chain P is called irreducible if for any two states $x, y \in \mathcal{X}$, there exists an integer t (possibly depending on x and y) such that $P^t(x, y) > 0$.

Intuitively, this means that it is possible to get from any state to any other state using only transitions of positive probability. This is verified in the current formulation of DDMs, indicating that DDMs satisfy the pre-requite to be an irreducible Markov chain.

Next, we recall the definition of period for a Markov chain.

Definition C.6. Let $\tau(x) := \{t \geq 1 : P^t(x, x) > 0\}$ be the set of times when it is possible for the chain to return to starting position x. The period of state x is defined to be the greatest common divisor of $\tau(x)$.

For an irreducible chain, the period of the chain is defined to be the period that is common to all states, and the chain is aperiodic if all states have period 1. Intuitively, the above definition and property match the actual formulation and implementations of DDMs, given the fact that plenty of existing DDMs [32, 4, 13, 39] propose an auxiliary loss to predict directly the denoised x_0 at arbitrary step.

Having introduced the above definitions, we are now ready to present the formal definition of mixing time in Markov chain studies.

Definition C.7. Definition of Mixing Time The mixing time is a parameter that measures the time required by a Markov chain for the distance to the stationary distribution to be small, following the definition below:

$$t_{mix}(\epsilon) := \min\{t : d(t) \leq \epsilon\} \text{ and } t_{mix} := t_{mix}(1/4). \tag{18}$$

In particular, taking $\epsilon = \frac{1}{4}$ above yields

$$d(lt_{mix}) \leq 2^{-l} \text{ and } t_{mix}(\epsilon) \leq \left\lceil log_2\, \epsilon^{-1} \right\rceil t_{mix}. \tag{19}$$

In addition to the initial definition of mixing time, we also need the background knowledge on the *time reversal* to search for the actual mixing step in a more practical way.

Definition C.8. For a distribution μ on a group G, the reversed distribution $\hat{\mu}$ is defined by $\hat{\mu}(g) := \mu(g^{-1})$ for all $g \in G$.

The time reversal is directly related to the two-direction design of DDMs, and ensures that the mixing step remains at **a fixed position** in two directions for both diffusion and generative processes. This property is also critical to better understand the DDMs, and provides theoretical justifications for us to search for the mixing step along the generative direction using the Gaussian radius estimation. In fact, the Gaussian radius estimation search method can only be valid and applied in the generative direction but not the inverse diffusion process. The reasons are discussed in the following section when we show more empirical results.

Lemma C.9. *Let P be the transition matrix of a random walk on a group G with increment distribution μ and let $\hat{P}$ be that of the walk on G with increment distribution $\hat{\mu}$. Let π be the uniform distribution on G. Then for any $t \geq 0$,*

$$||P^t(id, \cdot) - \pi||_{TV} = ||\hat{P}^t(id, \cdot) - \pi||_{TV}. \tag{20}$$

The lemma above implies the remark below, which will be used in our proof for the Property **??**.

Remark C.10. If t_{mix} is the mixing time of a random walk on a group and $\hat{t_{mix}}$ is the mixing time of the reversed walk, then $t_{mix} = \hat{t_{mix}}$.

We hereby finish introducing the necessary background on Markov mixing studies, and continue to a more detailed discussion of the mixing step problem of DDMs.

D More Discussion on Mixing Step

Inspired by the Markov mixing studies, we remark that the current formulation of DDMs satisfies several key assumptions as described in Appendix C, including most importantly, DDMs model an irreducible and aperiodic chain. Note that our current exploration and formulation for the mixing step of DDMs are not absolutely thorough and complete, which can be considered as an approximate and adapted version of the mathematical Markov mixing time.

D.1 Proof for Property of Mixing Step

We rewrite the Property of mixing step for DDMs here before going to the detailed proof.

Property D.1. *Under the total variation distance measure $|| \cdot ||_{TV}$, the mixing step t_m for a DDM with data dimensionality d is formed during training (i.e., irrelevant to the sampling methods). t_m is mainly related to the transition kernels and the stationary distribution (i.e., datasets), and less dependant on the dimensionality d.*

Proof. The proof for the above property consists of several steps.

Existence justification. Firstly, we have shown that DDMs model a group of chains that are irreducible and aperiodic, and thus the convergence theorem holds for DDMs. This fact establishes the theoretical foundation to find such a critical convergent step that theoretically characterizes the convergence of pre-trained DDMs.

Directions to approach. Secondly, we show that the mixing step is large and mostly dependent on the transition kernel. Here, we have to clarify the direction of the DDMs we are tackling. Fortunately, based on the time reversal from Lemma C.9 and Remark C.10, whichever direction gives the same mixing step, provides us with the flexibility to study either direction. However, in practice, the easiest way to approach the mixing step is to theoretically infer the transition kernel in the diffusion direction,

and then empirically search for it in the denoising direction. We will first provide the method and explain the reasons for such a design.

Theoretical based transition kernel study. We hereby restrict ourselves in considering the diffusion process. Given pre-trained DDMs, according to Lemma C.9, we have an irreducible transition matrix P on space $\mathcal{X}$. In the current scenario of diffusion direction from $\mathbf{x}_0$ to $\mathbf{x}_T$, the stationary distribution is the standard Gaussian $\mathcal{N}(0, \mathbf{I}_d)$ in ϵ_T. The transition matrix is a pre-defined Gaussian with known mean value and variance, thus we have

$$P = q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t I). \tag{21}$$

For an irreducible transition matrix P with stationary distribution π, define:

$$\sigma_t(x, y) := \frac{P^t(x, y)}{\pi(y)}, \tag{22}$$

with $\sigma_t(x, y) = \sigma_t(y, x)$ when P is reversible with respect to π. We also have:

$$< \sigma_t(x, \cdot), 1 >_\pi = \sum_y q_t(x, y)\pi(y) = 1. \tag{23}$$

Next, we have the definition of l^p-distance $d^{(p)}$ as:

$$d^{(p)}(t) := \max_{x\in\mathcal{X}} ||\sigma_t(x, \cdot) - 1||_p. \tag{24}$$

To replace the above notations with the notations from DDMs, we have:

$$d^{(1)}(t) := \max_{x\in\mathcal{X}} ||\sigma_t(x, y) - 1||_1, \tag{25}$$

and

$$\sigma_t(x, y) = \frac{P^t(x, y)}{\pi(y)} = \frac{x \sim \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)}{y \sim \mathcal{N}(0, I_d)}. \tag{26}$$

Based on the definition of mixing time in 18, we then have:

$$t^{(1)}_{mix}(\varepsilon) := \inf\{t \geq 0;\ d^{(1)}(t) \leq \varepsilon\}. \tag{27}$$

We take the value ε to be $\frac{1}{2}$, and thus arrive at:

$$t^{(1)}_{mix} := \inf\{t \geq 0; d^{(1)}(t) \leq \frac{1}{2}\}. \tag{28}$$

Now, we return back to Equation 26 and replace the Equation 28 with:

$$\max_{x\in\mathcal{X}} || \frac{x \sim \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)}{y \sim \mathcal{N}(0, I_d)} - 1|| \leq \frac{1}{2}. \tag{29}$$

By using the Proposition C.2, we can now substitute the above Equation 29 using the approximation as follows:

$$||x \sim \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)|| \leq 4. \tag{30}$$

This above gives an approximation of the transition kernel at the mixing step in the diffusion direction we would expect, with a radius change at approximately 4.

We observe that there is no explicit dependency on the dimensionality of the latent spaces, but directly related to the formulation of the transition kernel, which is mostly the Gaussian as used in existing DDMs implementations. In the meanwhile, we note the intermediate latent encodings x_t are actually dataset dependent. Therefore, we verify our claim that the mixing step is more dependent on the transition kernel and dataset. However, despite no explicit dependency between the mixing step and dimensionality, we empirically observe that the appearance of mixing step still differs in pre-trained diffusion models on different resolutions as in Tab. 2.

□

Interestingly, the above proof gives us a numerical approximation for the transition kernel when the mixing step appears, which is the radius variation at approximately 4. We hereby finish demonstrating the fact that the mixing step appears at around diffusion step $t = 500$ in our main paper. In the meanwhile, other related works [20, 21] report similar conclusions that editing on step 500 shows better empirical performance in different experimental settings.

Table 1: Gaussian radius estimation for empirical search of the mixing step for pre-trained DDPM on CelebA-64.

Steps	1000	900	800	700	600	**500**	400	300
$\mathbf{x}_T^s + p_s$	110.84	110.82	110.83	110.58	109.83	107.85	103.22	94.64
$\lvert \triangle r \rvert$	0.02	0.01	0.25	0.75	1.98	**4.63**	8.58	-
$\mathbf{x}_T^s + p_i$	110.88	110.86	110.84	110.63	109.89	107.86	103.10	94.45
$\lvert \triangle r \rvert$	0.02	0.02	0.21	0.74	2.03	**4.76**	8.65	-
$\mathbf{x}_T^i + p_s$	95.06	93.30	91.56	90.14	88.69	86.61	82.80	76.46
$\lvert \triangle r \rvert$	1.76	1.74	1.42	1.45	2.08	**3.81**	6.36	-
$\mathbf{x}_T^i + p_i$	95.06	93.34	91.61	90.16	88.75	86.57	82.87	76.53
$\lvert \triangle r \rvert$	1.72	1.73	1.45	1.41	2.18	**3.70**	6.34	-

Table 2: More results on Gaussian radius estimation for the empirical search of the mixing step for pre-trained DDMs

Model	Setting	1000	900	800	700	**600**	**500**	400
DDPM-CelebA-HQ-256	$\mathbf{x}_T^s + p_s$	443.42	443.36	443.30	442.49	440.16	432.55	416.77
	$\lvert \triangle r \rvert$	0.06	0.06	0.81	2.33	7.61	15.78	-
DDPM-CelebA-HQ-256	$\mathbf{x}_T^s + p_i$	443.40	443.29	443.06	442.22	439.44	431.66	413.96
	$\lvert \triangle r \rvert$	0.11	0.23	0.84	2.78	7.78	17.70	-
iDDPM+AFHQ-256	$\mathbf{x}_T^s + p_i$	443.34	443.21	442.84	441.72	439.31	429.16	408.69
	$\lvert \triangle r \rvert$	0.13	0.37	1.12	2.41	10.15	20.47	-

D.2 More Empirical Results on Mixing Step

For the empirical verification of the mixing step, we use a pre-trained DDPM model [15] on the CelebA dataset [24] with $3 \times 64 \times 64$ resolution. Therefore, for a standard Gaussian space in the dimensionality of $d = 3 \times 64 \times 64 = 12,288$, the expected Gaussian radius is $r = \sigma\sqrt{d} = 110.85$. The full radius estimation results are listed in Tab. 1, we also show the difference in Gaussian radius between consecutive 100 steps. Note we are slightly "abusing" the estimation results for steps after the mixing step, since the distributions of the latent spaces after ϵ_{t_m} are no longer considered as Gaussian, but rather converge to the actual data distributions in $\mathcal{X}$, therefore, estimating the Gaussian radius of those latent spaces are not theoretically sound. This also explains the reason why we do not report the numbers for step numbers less than 300. The above also explains our design to derive the theoretical proof in the diffusion direction, but proposes to empirically search for the mixing step via the denoising direction.

D.3 Connection with Existing Works

We notice that existing SOTA methods [20, 21] have proposed similar ideas in their works, by empirically exploring the diffusion steps that obtain better qualitative results. However, the mixing step has been studied as a hyper-parameter (*i.e.*, "return step" in [20] and "edit step" [21]) that influences the downstream qualities without formal definition. In this work, we formally define and introduce the concept of the mixing step, which originated from sound mathematical studies on the Markov mixing time, and provide a comprehensive perspective to re-think this "hyper-parameter". More excitingly, we discover that the theoretically driven deviation and our Gaussian radius estimation method come to a consistent conclusion and echo with previous literature in actual experimental tests.

E Boundary Search Discussion

In this section, we present more details about our proposed boundary search method, and discuss the connections between different latent space levels from the perspective of the Projection theorem.

E.1 Implementations

We use the linear SVM classifier for searching the semantic boundary. We implement the SVM via the sklearn python package with the number of parameters equal to the total dimensionality of the

latent spaces. For ϵ-space, the dimensionality $d_\epsilon = 3 \times 256 \times 256 = 196,608$. For the h-space, the dimensionality depends on the pre-trained DDMs architecture implementation for the U-Net [29]. In our experiments, we use the same level of latent spaces as in [21], which have a dimensionality of $d_h = 8 \times 8 \times 512 = 32,768$. In practice, we observe approximately 100 images are sufficient for finding an effective semantic boundary.

E.2 Projection Theorem

In theory, we expect the projected lower-dimensional subspace to preserve the same properties of its original higher-dimension space such as the projected distances between pairs of samples should have the same ordering in two spaces. In mathematics, we can ensure the validity of this projection design using the existing projection theorems.

Theorem E.1. ***Theorem of the Random Projection.*** *Let* $\mathbf{v}$ *be a fixed unit length vector in a* d*-dimensional space and let* W *be a random* k*-dimensional subspace. Let* $\mathbf{w}$ *be the projection of* $\mathbf{v}$ *onto* W*. For any* $0 \leq \epsilon \leq 1$*,* $Prob(||\mathbf{w}|^w - \frac{k}{d}| \geq \epsilon\frac{k}{d}) \leq 4e^{-\frac{k\epsilon^2}{64}}$.

One way to interpret the random projection theorem is that if one chooses a random k-dimensional subspace from a higher-dimensional space in d-dimension, then indeed all the projected distances in the k-dimensional space are approximately within a known scale factor of the distances in the d-dimensional space.

We present the projection theorem here to draw connections between the above boundary search and different operational latent space levels (*i.e.*, ϵ-space and h-space). As we describe in the main paper, the classification results from Tab. 3 show that even though the accuracy score is generally lower in ϵ-space, it does carry meaningful semantic boundaries. This above observation and claim differ from the previous literature [27], where the latent spaces of DDMs are considered to lack semantic meaning. In fact, given the recent study from [21], which first reveals the semantic behaviors of pre-trained DDMs in h-space, it provides evidence to imply that the same semantic meanings might also exist in the higher-dimensional ϵ-space. As h-space is a subspace of corresponding ϵ-space with higher dimensionality.

F Mixing Trajectory

We show the algorithm implementation for our proposed boundary-guided mixing trajectory under the conditional application scenario in Algo. 1. For the unconditional scenario, the only difference is that we can directly sample the latent encodings from the Gaussian distribution as the initial $\mathbf{x}_T$, and get the corresponding h-level latent encoding $\mathbf{h}_T$ from the given DDPM at T step.

G More Experimental Results

We present more experimental results and discussion in this section.

G.1 Pre-trained Models and Datasets

The pre-trained DDMs we use for experiments mainly include the DDPM [15] and the improved DDPM (iDDPM) [25]. The main difference between the original DDPM and the improved version lies within the fact that iDDPMs use a hybrid learning objective that obtains better log-likelihoods than directly optimizing it.

We conduct experiments on multiple datasets, which includes CelabA-64 [24], CelebA-HQ-256 [17], AFHQ-dog-256 [10], LSUN-church-256 [36], LSUN-bedroom-256 [36]. Different from existing works that usually pay little attention to the image resolutions in the experiments, the resolutions play an important role in our experiments since they define the actual dimensionality of the latent spaces for pre-trained DDMs. However, the 64^2 resolution model is mainly used in the high-dimensional analysis and interpolation observations, for the image editing and semantic control experiments, we use 256^2 as the default resolution for visualization quality.

Algorithm 1 Boundary Guided Mixing Trajectory (Conditional)

Input: input image $\mathbf{x}_0$, target boundaries $\mathbf{b}_\epsilon$ and $\mathbf{b}_h$ for the editing attribute m, pre-trained DDM p, inversion steps S_{inv}, denoising steps S_{gen}, mixing step t_m, user defined editing distance ζ_ϵ and ζ_h, and editing space steps K.
// Step 1: Inversion via DDIMs to get the latent encoding at t_m
Define $\{\tau_s\}_{s=1}^{S_{inv}}$ s.t. $\tau_1 = 0$, $\tau_{S_{inv}} = t_m$
for $s = 1, 2, ..., S_{inv} - 1$ **do**
  $\epsilon \leftarrow p(\mathbf{x}_{\tau_s}, \tau_s)$
  $\mathbf{x}_{\tau_{s+1}} = \sqrt{\alpha_{\tau_s}}\mathbf{x}_{\tau_s} + \sqrt{1-\alpha_{\tau_s}}\epsilon$
end for
$\mathbf{h}_{t_m} \leftarrow$ extract h feature map from ϵ
// Step 2: Boundary guidance
// Step 2.1: Define initial editing space in ϵ and h latent levels
$\{d_\epsilon^j\}^K$ s.t. $d_\epsilon^1 = -\zeta_\epsilon$, $d_\epsilon^K = \zeta_\epsilon$
$\{d_h^j\}^K$ s.t. $d_h^1 = -\zeta_h$, $d_h^K = \zeta_h$
// Step 2.2: Compute projection distance to the boundaries
$\{d_{p,\epsilon}\} = \{d_\epsilon\} - \mathbf{b}_\epsilon^T \mathbf{x}_{t_m}$
$\{d_{p,h}\} = \{d_h\} - \mathbf{b}_h^T \mathbf{h}_{t_m}$
// Step3: Denoising with mixing trajectory
for k = 1, 2, ..., K **do**
  $\mathbf{x}'_{t_m} = \mathbf{x}_{t_m} + d_\epsilon^k \mathbf{b}_\epsilon$
  $\mathbf{h}'_{t_m} = \mathbf{h}_{t_m} + d_h^k \mathbf{b}_h$
  $\mathbf{x}_s \leftarrow \mathbf{x}'_{t_m}$
  $\mathbf{h}_s \leftarrow \mathbf{h}'_{t_m}$
  for $s = S_{gen}, S_{gen} - 1, ..., 2$ **do**
    $\epsilon \leftarrow p_s(\mathbf{x}_s, \mathbf{h}_s, s)$
    $z \sim \mathcal{N}(0, I_d)$
    $\mathbf{x}_{s-1} = \sqrt{\alpha_{s-1}}(\frac{\mathbf{x}_s - \sqrt{1-\alpha_s}\epsilon}{\sqrt{\alpha_s}}) + \sqrt{1-\alpha_{s-1}-\sigma^2}\epsilon + \sigma_s z$
  end for
end for

G.2 Semantic Boundary Validation

We search the semantic boundaries via linear classifiers on both ϵ-space and *h*-space using 100 images, and we show the semantic behaviors via the testing classification accuracy on different attributes in Tab. 3. We observe from the classification results that the boundaries in *h*-space are in general better defined compared to the ϵ-space, which is consistent with previous findings from [21]. Notably, for certain attributes such as *glass* and *mustache*, both space levels perform well in defining the boundaries, which implies and aligns with our empirical finding that guidance on both levels of latent spaces helps for more effective semantic control.

Table 3: Classification accuracy on separation boundaries in different latent spaces at the mixing step.

Latent space	Smile	Glass	Age	Mustache
ϵ	0.86	0.95	0.87	0.96
h	**0.98**	0.95	**0.93**	0.96

G.3 User Study

As subjective evaluations, we conduct user study to compare our proposed method with Asyrp [21] and DiffusionCLIP [20] on CelebA-HQ-256 [24]. We use the official codebases from previous works and follow the exact default commands, using the *smile* attribute as the editing target, either to add or remove smiles from 100 raw images that are randomly selected from the dataset.

We interviewed 20 human evaluators and asked similar questions as in previous works. Specifically, we asked the evaluators to pick the best edited result in terms of two main aspects: 1) General quality: which image quality do you think is the best? (clear, fidelity, photorealistic) 2) Attribute: which image

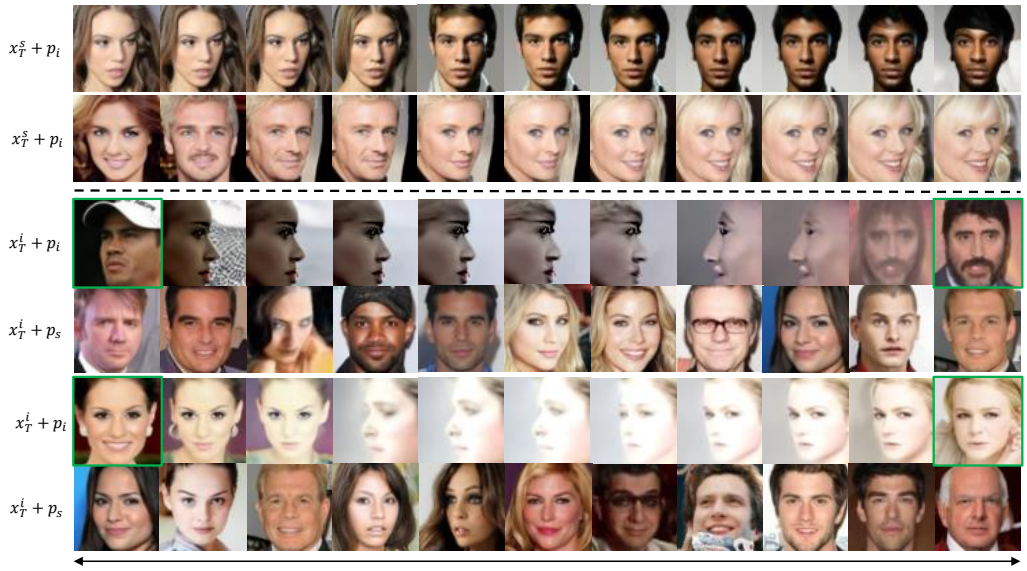

Figure 2: More interpolation results from different combinations of latent encoding sources and sampling methods on CelebA-64.

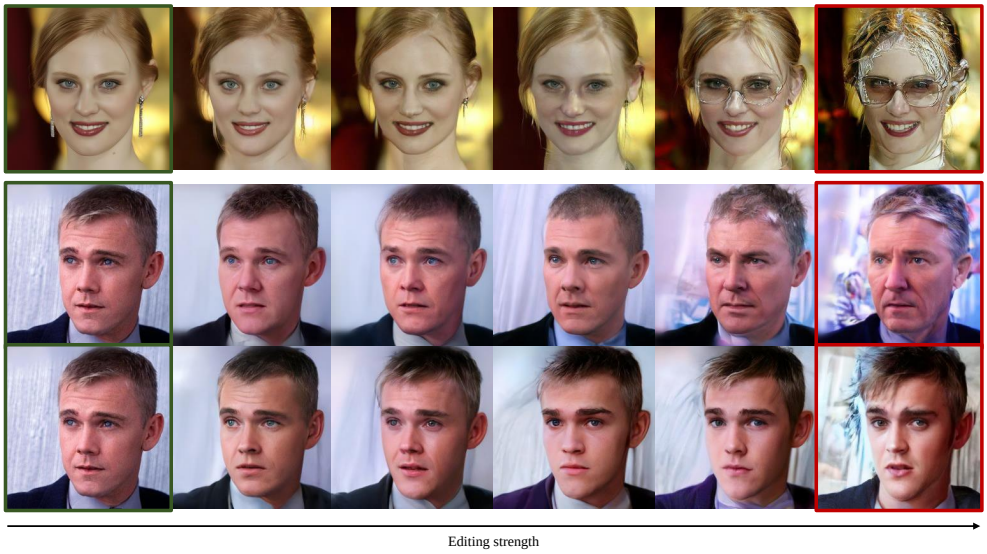

Figure 3: More qualitative results for editing strength modification. We use the CelabA-HQ-256 dataset and the attributes *glass* and *age* as examples. In particular, we also show samples with distortions and lower quality when the editing distance becomes too large. The optimal editing distance range is also related to the properties of the high-dimensional spaces.

do you think achieve the best attribute editing effect? (natural, identity preservation with respect to the given raw image)

More **non-cherry picky** editing results from three different methods are included in Fig. 1.

Figure 4: More qualitative results for text-based conditional editing on the LSUN-Bedroom-256 dataset.

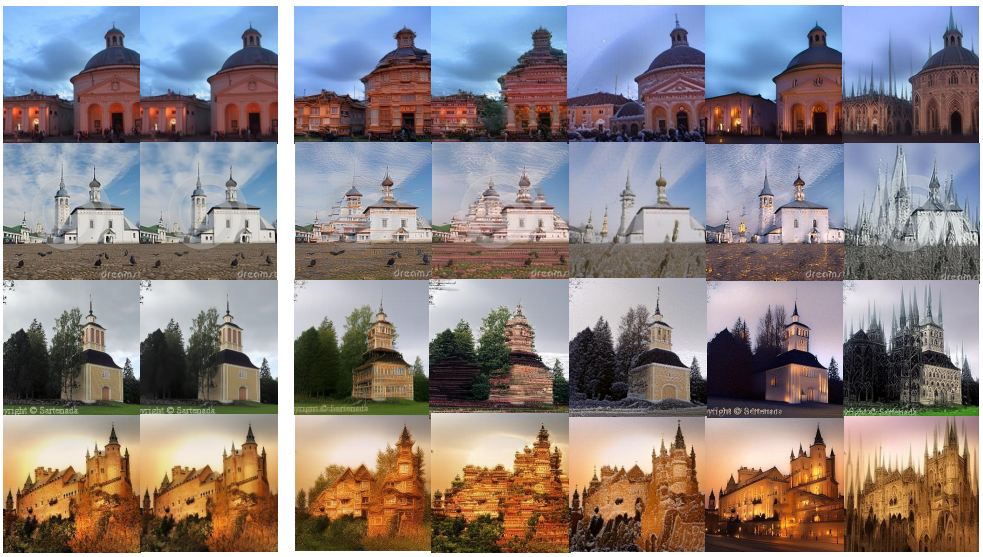

Figure 5: More qualitative results for text-based conditional editing on the LSUN-Church-256 dataset.

G.4 More Qualitative Results

H Further Discussion

H.1 Limitations

We discuss several limitations of our current work, which also bring insights for future research directions.

Firstly, while our work well preserves the original properties and potential of pre-train DDMs, we have not yet well tested its ability to achieve multiple attributes of semantic editing at one time. However, we believe this is also feasible by leveraging the technique of multi-hyperplane projections.

Figure 6: More qualitative results for unconditional semantic control on AFHQ-Dog-256 and CelabA-HQ-256 datasets with *smile* semantic.

At the same, we find it rather difficult to apply our current boundary-guided mixing trajectory to unseen domain transfer problems without changing any parameters or learning extra neural network modules. Despite we have shown some qualitative results for unseen style transfer with reasonable quality for the transfer such as "*dog-to-zombie*" and "*dog-to-fox*", it is more challenging for pre-trained DDMs on AFHQ-Dog to capture a clear boundary and find the appropriate trajectory to generate a human face. However, we are still optimistic about this direction, given the fact that Aysrp [21] has shown the ability to perform unseen domain transfer tasks well using frozen pre-trained DDMs, the remaining challenge is about finding a more sophisticated way to do improved optimization.

H.2 Time and Resource

Compared to previous works that require either fine-tuning the pre-trained DDMs [20], or learning an extra editing network [21], our approach seeks to find an existing semantic boundary with frozen DDMs without learning any additional extra neural networks. In practice, the hyperplanes are found via linear SVMs [14], with almost negligible learning time of about 1 second on a single RTX3090 GPU. The number of parameters in an SVM classifier is the same as the dimensionality.

For the inference, the time cost remains at the same level as other SOTA methods. Specifically, by using the skipping step techniques, we can already generate high-quality denoised images using approximately 40-100 steps, which take from 1.682 - 13.272 seconds, respectively on a single RTX-3090 GPU.

H.3 Broader Impact

We discuss the broader impact of this work. Firstly, the primary goal of this work is not to create new generative models or generate synthetic data, but to explore the potential of the current generative models for better usage. To do so, we also propose a new perspective to better understand and interpret the DDMs, which is the analysis of high-dimensional latent space behaviors using the theoretical tools from mathematics and statistics. In the meanwhile, during the process of exploring and separating the semantic boundary, we leverage the current popular cross-modality generative models to synthesize images with a text prompt. However, all the generated images are only used for boundary detection.

We believe our work brings valuable insights to the research community in terms of a better understanding and further exploration via training-free methods to apply diffusion generative models.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *ICCV*, 2019.

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *CVPR*, 2020.

[3] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *ICCV*, 2021.

[4] Jacob Austin, Daniel Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *NeurIPS*, 2021.

[5] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Towards open-set identity preserving face synthesis. In *CVPR*, 2018.

[6] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Inverting layers of a large generator. In *ICLR Workshop*, 2019.

[7] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *ICCV*, 2019.

[8] Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of data science*. Cambridge University Press, 2020.

[9] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

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

[11] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE TNNLS*, 2018.

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

[13] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, 2022.

[14] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 1998.

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

[16] Minyoung Huh, Richard Zhang, Jun-Yan Zhu, Sylvain Paris, and Aaron Hertzmann. Transforming and projecting images into class-conditional generative networks. In *ECCV*. Springer, 2020.

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

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

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

[20] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *CVPR*, 2022.

[21] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. In *ICLR*, 2023.

[22] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. Fader networks: Manipulating images by sliding attributes. *NIPS*, 2017.

[23] David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

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

[25] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*. PMLR, 2021.

[26] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017.

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

[28] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *CVPR*, 2021.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and*

computer-assisted intervention. Springer, 2015.
[30] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020.
[31] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*. PMLR, 2015.
[32] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021.
[33] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *CVPR*, 2017.
[34] Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Weiming Zhang, Lu Yuan, Gang Hua, and Nenghai Yu. E2style: Improve the efficiency and effectiveness of stylegan inversion. *IEEE TIP*, 2022.
[35] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *IEEE TPAMI*, 2022.
[36] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.
[37] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *ECCV*. Springer, 2020.
[38] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *ECCV*. Springer, 2016.
[39] Ye Zhu, Yu Wu, Kyle Olszewski, Jian Ren, Sergey Tulyakov, and Yan Yan. Discrete contrastive diffusion for cross-modal and conditional generation. In *ICLR*, 2023.