Exploring the Design Space of Diffusion Bridge Models

Shaorong Zhang, Yuanbin Cheng, Greg Ver Steeg
Unversity of California Riverside
{szhan311, ychen871, gregoryv}@ucr.edu

Abstract

Diffusion bridge models and stochastic interpolants enable high-quality image-to-image (I2I) translation by creating paths between distributions in pixel space. However, recent diffusion bridge models excel in image translation but suffer from restricted design flexibility and complicated hyperparameter tuning, whereas Stochastic Interpolants offer greater flexibility but lack essential refinements. We show that these complementary strengths can be unified by interpreting all existing methods within a single SI-based framework. In this work, we unify and expand the space of bridge models by extending Stochastic Interpolants (SIs) with preconditioning, endpoint conditioning, and an optimized sampling algorithm. These enhancements expand the design space of diffusion bridge models, leading to state-of-the-art performance in both image quality and sampling efficiency across diverse I2I tasks. Furthermore, we identify and address a previously overlooked issue of low sample diversity under fixed conditions. We introduce a quantitative analysis for output diversity and demonstrate how we can modify the base distribution for further improvements. Code is available at `https://github.com/szhan311/ECSI`.

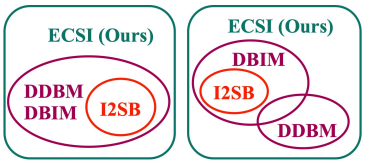
1 Introduction

Denoising Diffusion Models (DDMs) and flow matching create a stochastic process to transition Gaussian noise into a target distribution [33, 14, 34, 19]. Building upon this, diffusion bridge-based models (DBMs) have been developed to transport between two arbitrary distributions, π_T and π_0, including I2SB [21], DSBM [39], DDBM [18], DBIM [42], Bridge Matching [28]. DBMs achieve superior image quality in I2I translation compared to DDMs [18, 21, 2], primarily because the distance between source and target image distributions is typically smaller than that between Gaussian and target distributions.

While DBMs like DDBM [39], DBIM [42], and I2SB [21] achieve state-of-the-art FID scores in image-to-image translation, they suffer from limited design flexibility, constrained bridge path formulations, and complex parameter tuning. In contrast, Stochastic Interpolants (SIs) [1, 2] offer a simpler and more flexible framework, but they have yet to integrate practical advances from recent diffusion bridge models, such as preconditioning. Besides, SIs require training two separate models, unlike the more efficient single-model setup in DDBM. Table 1 summarizes the key characteristics of these methods, highlighting that their complementary strengths had not yet been unified.

Another overlooked issue stemming from restrictive design choices in previous bridge models is the lack of diversity in outputs. While some image translation tasks are one-to-one, we find that in one-to-many translation tasks, like black and white edges to color images, previous methods produce limited variation in colors and textures. We refer to this as the *conditional diversity* problem and show that our approach leads to significant improvements.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

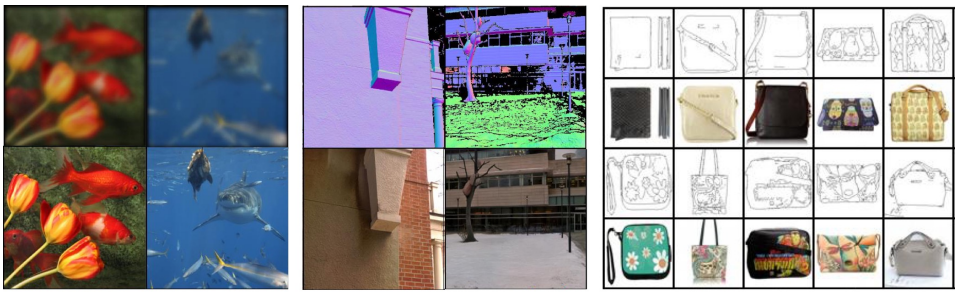


(a) Bridge paths (b) Samplers

Figure 1: The design space of bridge paths and samplers.

	DDBM	DBIM	DSBM	SI	ECSI (ours)
Endpoint conditioning	✓	✓	✗	✗	✓
Uncoupled parameters	✗	✗	✗	✓	✓
Extensive bridge paths	✗	✗	✓	✓	✓
Extensive samplers	✗	✗	✗	✓	✓
Preconditioning	✓	✓	✗	✗	✓
Modified base density	✗	✗	✗	✗	✓

Table 1: Characteristics of different bridge models.



Deblurring Depth to RGB Edges to Hangbags

Figure 2: Samples for I2I translation with our ECSI models: Deblurring, Depth-RGB, and Edges to Handbags. For each pair of images, we show the input image (upper) and the output image (bottom).

Our main contributions are as follows:

- We propose **Endpoint-Conditioned Stochastic Interpolants** (ECSI), which extend stochastic interpolants by incorporating endpoint conditioning and preconditioning. Previous bridge methods artificially coupled unrelated aspects of the transition kernel. ECSI introduces a decoupled parametrization that expands and simplifies the design space for bridge paths and samplers. To further improve sampling quality and efficiency, we develop a novel noise control scheme and an efficient sampling algorithm.
- We identify a previously overlooked issue: the low diversity of outputs conditioned on fixed source images. To address this, we propose modifying the base distribution. Furthermore, to quantitatively evaluate conditional output diversity, we introduce a new metric—Average Feature Diversity (AFD).
- Experimental results demonstrate our model's state-of-the-art performance in both image quality and sampling speed across various I2I tasks, including deblurring, edges-to-handbags translation, and depth-to-RGB conversion. Notably, for handbag generation, our approach yields significantly more diverse outputs with varied colors and textures.

2 Background

Notations Let π_T, π_0, and π_{0T} represent the base distribution, the target distribution, and the joint distribution of them respectively. π_{cond} and π_{data} represent the distributions of the input and output data. Let p be the distribution of a diffusion process; we denote its marginal distribution at time t by p_t, the conditional distribution at time t given the state at time s by $p_{t|s}$, and the distribution at time t given the states at times 0 and T by $p_{t|0,T}$, i.e., the transition kernel of a bridge.

2.1 Denoising Diffusion Bridge Models

DDBMs [18] extend diffusion models to translate between two arbitrary distributions π_0 and π_T given samples from them. Consider a reference process given by:

$$dX_t = f_t X_t dt + g_t dW_t, \tag{1}$$

whose transition kernel is given by $q_{t|0}(x_t|x_0) = \mathcal{N}(x_t; a_t x_0, \sigma_t^2 \mathbb{I})$. This process can be conditioned (or "pinned") at both an initial point x_0 and a terminal point x_T to construct a diffusion bridge. Under mild assumptions, the pinned process is given by Doob's h-transform [29]:

$$\mathrm{d}X_t = \{f_t X_t + g_t^2 \nabla_{X_t} \log p_{T|t}(x_T|X_t)\}\mathrm{d}t + g_t \mathrm{d}W_t \tag{2}$$

where $\nabla_{X_t} \log p_{T|t}(x_T \mid X_t) = \frac{(a_t/a_T)x_T - X_t}{\sigma_t^2(\mathrm{SNR}_t/\mathrm{SNR}_T - 1)}$ and $\mathrm{SNR}_t := a_t^2/\sigma_t^2$ [18]. Eq. (2) is a stochastic process that transport from $p_0 = \pi_0$ and $p_t = \pi_t$, which is a valid bridge process. To sample from the conditional distribution $p(x_0|x_T)$, we can solve the reverse SDE or probability flow ODE from $t = T$ to $t = 0$ [18]:

$$\mathrm{d}X_t = \{f_t X_t + g_t^2(s - h)\}\mathrm{d}t + g_t \mathrm{d}W_t, \tag{3}$$

$$\mathrm{d}X_t = \{f_t X_t + g_t^2(s - \frac{1}{2}h)\}\mathrm{d}t, \tag{4}$$

where $X_T = x_T$, $s = \nabla_{X_t} \log p_{T|t}(x_T|X_t)$, $h = \nabla_{X_t} \log p_{t|T}(X_t|x_T)$. Generally, the score $\nabla_{x_t} \log p_{t|T}(x_t|x_T)$ in Eqs. (3) and (4) is intractable. However, it can be effectively estimated by denoising bridge score matching. Let $(x_0, x_T) \sim \pi_{0,T}(x_0, x_T)$, $x_t \sim p_{t|0,T}(x_t|x_0, x_T)$, $t \sim \mathcal{U}(0, T)$, and $\omega(t)$ be non-zero loss weighting term of any choice, then the score $\nabla_{x_t} \log p_{T|t}(x_T|x_t)$ can be approximated by a neural network $s_\theta(x_t, x_T, t)$ with denoising bridge score matching objective [18]:

$$\mathcal{L}(\theta) = \mathbb{E}\left[w(t)\|s_\theta(X_t, x_T, t) - \nabla_{x_t} \log p_{t|0,T}(X_t \mid x_0, x_T)\|^2\right]. \tag{5}$$

where $\mathbb{E}$ dentotes expectation over $x_t \sim p_{t|0,T}(x_t, x_0)$, $(x_0, x_T) \sim \pi_{0,T}$, $t \sim \mathcal{U}(0, T)$.

2.2 Diffusion Bridge Implicit Models

The transition kernel of the bridge process in Eq. (2) is given by [18, 42]:

$$p(x_t|x_0, x_T) = \mathcal{N}(x_t; \alpha_t x_0 + \beta_t x_T, \gamma_t^2 \mathbb{I}) \tag{6}$$

where $\alpha_t = a_t(1 - \frac{\mathrm{SNR}_T}{\mathrm{SNR}_t})$, $\beta_t = \frac{a_t}{a_T}\frac{\mathrm{SNR}_T}{\mathrm{SNR}_t}$, $\gamma_t^2 = \sigma_t^2(1 - \frac{\mathrm{SNR}_T}{\mathrm{SNR}_t})$. Suppose we sample in reverse time on the discretized timesteps $0 = t_0 < t_1 < \cdots t_{N-1} < t_N = T$. Then we can sample x_0 by the initial value x_T and the updating rule:

$$x_{t_n} = \alpha_{t_n} x_T + \beta_{t_n} \hat{x}_0^\theta + \sqrt{\gamma_{t_n}^2 - \rho_{t_n}^2}\frac{x_{t_{n+1}} - \alpha_{t_{n+1}} x_T - \beta_{t_{n+1}} \hat{x}_0^\theta}{\gamma_{t_{n+1}}} + \rho_{t_n}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbb{I}). \tag{7}$$

where $\hat{x}_0^\theta(x_t, x_T, t)$ has the relation with the score function:

$$s_\theta(x_t, x_T, t) = -\frac{x_t - \alpha_t x_T - \beta_t \hat{x}_0^\theta(x_t, x_T, t)}{\gamma_t^2} \tag{8}$$

3 Have the bridge paths been fully explored?

Given the forward process defined in Eq. (1), diffusion bridge models [18, 42, 12, 8] utilize Doob's h-transform to construct a corresponding bridge process (Eq. (2)). While the resulting process effectively bridges the initial distribution π_T and the target distribution π_0, such diffusion bridge approaches exhibit several limitations.

- **Parameter coupling.** Notice that the parameters a_t and σ_t are convolved in the transition kernel (Eq. (6)). Such coupling is unnecessary and decoupling those parameters is helpful for searching the 'best' bridge path.

- **Limited design space.** Despite Eq. (2) provides an infinite number of bridge paths by tuning a_t and σ_t, but the space of bridge paths is still artificially restricted.

In contrast, the stochastic interpolants [1] framework allows a larger design space of bridge path with more decoupled parameters. Specifically, stochastic interpolants build a bridge path directly via the flow map:

$$\phi_t = \alpha_t x_0 + \beta_t x_T + \gamma_t z \tag{9}$$

where $z \sim \mathcal{N}(0, \mathbb{I})$. Eq. (9) builds a transport with π_0 and π_T as boundary conditions if the kernel parameters satisfy [1]:

- $\alpha_0 = \beta_T = 1$ and $\alpha_T = \beta_0 = \gamma_0 = \gamma_1 = 0$;
- $\alpha_t, \beta_t, \gamma_t > 0$ for $t \in (0, T)$.

The transition kernel of the stochastic interpolants in Eq. (9) is a Gaussian distribution: $\mathcal{N}(x_t; \alpha_t x_0 + \beta_t x_T, \gamma_t^2 \mathbb{I})$. Unlike DDBM, which is parameterized by only two variables a_t and σ_t, stochastic interpolants introduce decoupled parameters α_t, β_t, and γ_t, offering a more flexible and expressive design space for constructing bridge paths.

A detailed discussion on the rationale behind the choices of α_t, β_t, and and an ablation study on the shape of γ_t is provided in App. E. Notably, the DDBM-VP and DDBM-VE models presented in [18] can be considered as special cases by choosing different α_t, β_t, and γ_t, see App. D for more details. In the experiments, we limit the scope to linear transition kernels and set $T = 1$, i.e., $p_{t|0,T}(x_t|x_0, x_T) = \mathcal{N}(x_t; (1-t)x_0 + t x_1, 4\gamma_{\max}^2 t(1-t)\mathbb{I})$.

Stochastic interpolants expands the space of bridge paths and leads to decoupled parameters compared to DDBM and DBIM.

4 Has the sampler space been fully explored?

For diffusion models, EDM [17] demonstrated that the design of training and sampling schemes could be decoupled to significantly improve results. We now explore whether a similar decoupling is possible for bridge models, and what freedom we have to improve sampling quality with a given trained model.

4.1 Endpoint-Conditioning for Stochastic Interpolants (ECSI)

Given transition kernel $p_{t|0,T}(x_t \mid x_0, x_T) = \mathcal{N}(x_t; \alpha_t x_0 + \beta_t x_T; \gamma_t \mathbb{I})$, we can identify the training objective 11, reverse sampling SDEs (Eq. (10)), as demonstrated in Proposition 4.1, see App. C for the proof.

Proposition 4.1 (Endpoint-conditioned Stochastic Interpolants)**.** *Suppose the transition kernel of a diffusion bridge process is given by $p_{t|0,T}(x_t \mid x_0, x_T) = \mathcal{N}(x_t; \alpha_t x_0 + \beta_t x_T, \gamma_t^2 \mathbb{I})$, then the evolution of conditional probability $q_t(X_t|x_T)$ is given by the SDE:*

$$dX_t = b(t, X_t, x_T)dt + \sqrt{2\epsilon_t} dW_t, \tag{10}$$

where $b(t, x_t, x_T) = \dot{\alpha}_t \hat{x}_0 + \dot{\beta}_t x_T + (\dot{\gamma}_t + \frac{\epsilon_t}{\gamma_t})\hat{z}_t$, $\hat{x}_0 = \mathbb{E}[x_0 \mid x_t, x_T]$, $\hat{z}_t =: (x_t - \alpha_t \hat{x}_0 - \beta_t x_T)/\gamma_t$. Besides, $\hat{x}_0$ can be approximated by neural networks $\hat{x}_0^\theta$ by minimizing a regression objective with the observed x_0, x_T as targets,

$$\mathcal{L}_0[\hat{x}_0^\theta] = \int_0^T \mathbb{E}[\|\hat{x}_0^\theta(t, x_t, x_T) - x_0\|_2^2] dt \tag{11}$$

where $\mathbb{E}$ denotes an expectation over $(x_0, x_T) \sim \pi(x_0, x_T)$ and $x_t \sim p_t(x_t \mid x_0, x_T)$.

Relation to Stochastic Interpolants (SI). Both SI and ECSI in Prop. 4.1 can be seen as special cases of Conditioned SI. A key advantage of ECSI is its efficiency: while SI need to estimate two terms: $\mathbb{E}[x_0 \mid x_0]$ and $\mathbb{E}[x_1 \mid x_t]$, ECSI only estimate $\mathbb{E}[x_0 \mid x_t, x_1]$. A detailed comparison was demonstrated in App. B.

For training, we found that we could define an expanded space of bridge paths in terms of $\alpha_t, \beta_t, \gamma_t$, where γ_t apparently controlled the *stochasticity* of the path. For sampling, we see from the proposition above that the sampling design space is expanded even further, as the sampling dynamics depend on $\alpha_t, \beta_t, \gamma_t$ and ϵ_t, where ϵ_t appears as an additional degree of freedom to control stochasticity.

Training. Eq. (11) provides the training objective of the denoiser $\hat{x}_0^\theta(t, x_t, x_T)$. In the implementation, we include additional preconditioning as DDBM [18] and DBIM [42], see App. G for more details.

Sampling. We can generate samples from the conditional distribution $q_{0|T}(x_0 \mid x_T)$ by solving the stochastic differential equation in Eq. (10) from $t = T$ to $t = 0$.

4.2 Existing samplers are a strict subset of ECSI samplers

We now show that existing samplers implement a strict subset of the ECSI samplers, see Figure 1.

DDBM sampler. When $\epsilon_t = 0$, Eq. (10) reduces to a deterministic ODE. Setting $\epsilon_t = \gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2$ recovers the sampling SDE used in DDBM [18]. However, DDBM only provides a single reverse SDE and a single corresponding reverse ODE; it does not explore alternative choices of ϵ_t.

DBIM sampler. For small enough Δt and $\gamma_{t-\Delta t}^2 - 2\epsilon_t\Delta t > 0$, the sampling SDE can be discretized as:

$$x_{t-\Delta t} \approx \alpha_{t-\Delta t}\hat{x}_0 + \beta_{t-\Delta t}x_T + \tilde{z} \tag{12}$$

where $\bar{z}_t \sim \mathcal{N}(0, \mathbb{I})$, $\tilde{z} = \sqrt{\gamma_{t-\Delta t}^2 - 2\epsilon_t\Delta t}\hat{z}_t + \sqrt{2\epsilon_t\Delta t}\bar{z}_t$. Eq. (12) recover the DBIM sampler. Note that the condition $\gamma_{t-\Delta t}^2 - 2\epsilon_t\Delta t > 0$ limits the design space of samplers. For example, our best result in the experiments is achieved by setting $\alpha_t = 1 - t$, $\gamma_t = \frac{\gamma_{\max}^2}{4}t(1-t)$ and $\epsilon_t = \gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2$, DBIM sampler fails under this setting since $\gamma_{t-\Delta t}^2 - 2\epsilon_t\Delta t > 0$ cannot be guaranteed all the time.

I^{2}SB sampler. When $2\epsilon_t\Delta t = \gamma_{t-\Delta t}^2 - \beta_{t-\Delta t}^2\gamma_t^2/\beta_t^2$, the coefficient of x_T in Eq. (12) vanishes. This special case corresponds to the Markovian bridge introduced in [42], and notably allows us to recover the sampling procedure of I2SB [21]. We provide a detailed derivation of this connection in Appendix D. The design space of the I^{2}SB sampler is also limited, as it can be interpreted as a special case of the DBIM sampler.

Endpoint-Conditioned Stochastic Interpolants (Prop. 4.1) identify a class of sampling SDEs that share the same marginal distribution, but offer greater flexibility and a broader design space for sampler construction compared to DDBM, DBIM, and I2SB.

4.3 Our implementation

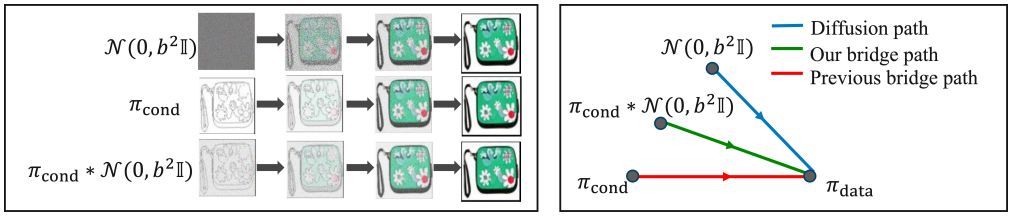
Our sampler based on Euler's discretization of the sampling SDE in Eq. (10):

$$x_{t-\Delta t} \approx x_t - b(t, x_t, x_T)\Delta t + \sqrt{2\epsilon_t\Delta t}\bar{z}_t, \tag{13}$$

We set $\epsilon_t = \eta(\gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2)$, where $\eta \in (0, 1)$ is an interpolation parameter. This formulation provides continuous control over the sampling process, ranging from purely deterministic ODE sampling ($\eta = 0$) to fully stochastic SDE sampling ($\eta = 1$). In our implementation, we let $\epsilon_t = 0$ for the last two steps, Eq. (12) gets reduced to: $x_{t-\Delta t} \approx \alpha_{t-\Delta t}\hat{x}_0 + \beta_{t-\Delta t}x_T + \gamma_{t-\Delta t}\hat{z}_t$. For other steps, we apply Eq. (13) and let $\epsilon_t = \eta(\gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2)$, where η is a constant. Putting all ingredients together leads to our sampler outlined in Algorithm 1.

Algorithm 1 ECSI Sampler

Input: $D_\theta(x_t, x_T, t)$, timesteps $\{t_j\}_{j=0}^N$, distribution π_{cond}, schedule $\alpha_t, \beta_t, \gamma_t, \epsilon_t, b$
Sample $x_T \sim \pi_{\text{cond}}$, $n_0 \sim \mathcal{N}(0, b^2\mathbb{I})$
$x_N = x_T + n_0$
for $i = N$ **to** 1 **do**
  $\hat{x}_0 = D_\theta(x_i, x_T, t_i), \quad \hat{z}_i = (x_i - \alpha_{t_i}\hat{x}_0 - \beta_{t_i}x_N)/\gamma_{t_i}$
  if $i \geq 2$ **then**
    Sample $\bar{z}_i \sim \mathcal{N}(0, \mathbb{I})$
    $d_i = \dot{\alpha}_{t_i}\hat{x}_0 + \dot{\beta}_{t_i}x_N + (\dot{\gamma}_{t_i} + \epsilon_{t_i}/\gamma_{t_i})\hat{z}_i$
    $x_{i-1} = x_i + d_i(t_i - t_{i-1}) + \sqrt{2\epsilon_{t_i}(t_i - t_{i-1})}\,\bar{z}_i$
  else
    $x_{i-1} = \alpha_{t_{i-1}}\hat{x}_0 + \beta_{t_{i-1}}x_N + \gamma_{t_{i-1}}\hat{z}_i$
  end if
end for



(a) Bridge paths in state space (b) Bridge paths in density space

Figure 3: Modifying the base distribution corresponds to a lossy compression of the input that leads to a 'trade-off' between unconditional diffusion and diffusion bridge models.

5 Is there any benefit to modifying the starting point of a bridge?

We expanded the paths in distribution space connecting a base and target distribution, but so far left the endpoints fixed. While the target distribution should remain fixed, we could, in principle, modify the base distribution. At first glance this seems counter-intuitive - because of the data processing inequality we can only lose information about the target by modifying the base distribution. Hence, this angle has not been explored in the bridge literature. However, we found a surprising result - modifying the base distribution can help significantly. The situation is analogous to the benefits of lossy compression in VAEs [6]. Information in the base distribution is not necessarily helpful, so by modifying the base distribution (which destroys some information) the model can align better with natural factors of variation.

5.1 Low conditional diversity in one-to-many translations

In our experiments (see Sec. 6), we observe that existing diffusion bridge models tend to produce low-diversity outputs under fixed conditioning. For instance, when generating handbags from a single edge map, the model is expected to produce varied outputs in terms of color, texture, and fine details. However, we find that current bridge models generate visually similar images across different sampling runs, despite the injection of different noise realizations during the diffusion process.

To address the issue of low output diversity, we propose modifying the base distribution used in the bridge model. Prior works [18, 1] typically treat the base distribution π_T as equivalent to the input data distribution, denoted π_{cond}. In contrast, our approach introduces a controlled perturbation by redefining the base distribution as $\pi_T = \pi_{\text{cond}} * \mathcal{N}(0, b^2\mathbb{I})$, where b is a constant that governs the magnitude of noise added to the input distribution. This modification enables greater diversity in the generated outputs while maintaining conditional alignment.

Intuitively, this modification can be interpreted as a trade-off between standard diffusion models and traditional diffusion bridge models. As illustrated in Fig. 3, diffusion models typically generate samples starting from pure Gaussian noise, while diffusion bridge models begin sampling from fully conditioned inputs, such as edge maps. Our approach introduces an intermediate regime by

Table 2: Validation of our sampler via DDBM pretrained VP model (Evaluated by FID), where $\epsilon_t = 0.3(\gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2)$.

Sampler	Edges→Handbags (64×64)			DIODE-Outdoor (256×256)		
	NFE=5	NFE=10	NFE=20	NFE=5	NFE=10	NFE=20
DDBM [18]	317.22	137.15	46.74	328.33	151.93	41.03
DBIM [42]	3.60	2.46	1.74	14.25	7.98	4.99
ECSI (Ours)	**2.36**	**2.25**	**1.53**	**10.87**	**6.83**	**4.12**

sampling from noisy conditioned inputs, thereby blending the benefits of both paradigms—preserving conditional guidance while enhancing output diversity.

Modifying the base distribution with lossy compression can significantly improve the conditional diversity of the generated images.

5.2 How to measure the conditional diversity?

While existing metrics like FID implicitly capture the *unconditional* diversity of generated images, we need to capture the diversity of outputs (e.g. color images) for a single input image (a black and white edge map). To measure the conditional diversity, we will adopt Vendi Score (VS) [11] as a metric. Besides, We propose the Average Feature Distance (AFD) metric to quantify the conditional diversity among generated images. Initially, we select a group of source images $\{x_T^{(i)}\}_{i=1}^M$. For each $x_T^{(i)}$, we then generate L distinct target samples. The j-th generated sample corresponding to the i-th source image is denoted by y_{ij}. Then the AFD is calculated as follows:

$$\text{AFD} = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{L^2 - L}\sum_{k,l=1,k\neq l}^{L}\|F(y_{ik}) - F(y_{il})\| \tag{14}$$

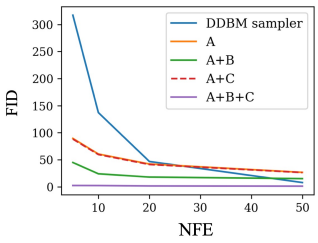
where $F(\cdot)$ is a function that extracts the features of images, and $\|\cdot\|$ represents Euclidean norm. Intuitively, a larger AFD indicates the better conditional diversity. Here, $F(x)$ can be x to evaluate the diversity directly in the pixel space. Alternatively, $F(\cdot)$ can be defined using the Inception-V3 model to assess the diversity in the latent space. In our experiments, we use AFD in latent space. Furthermore, we provide additional justification for the validity of our proposed metric in App. A.

A comparison between AFD and VS. Both AFD and the VS quantify diversity in the feature space of images, using features extracted from the Inception-V3 model. AFD measures the average pairwise Euclidean distance between feature vectors, making it sensitive to outliers. In contrast, the Vendi Score evaluates diversity by computing the effective number of unique feature patterns, based on the eigenvalues of the similarity matrix, emphasizing the overall structural diversity of the feature set. These metrics are complementary, capturing different aspects of diversity.

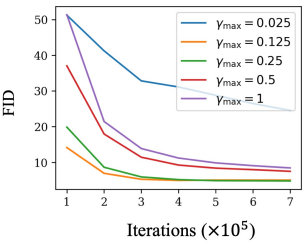
6 Experiments

In this section, we demonstrate how greatly expanding the space of bridge paths with ECSI leads to significantly improved performance for I2I translation tasks, in terms of sample efficiency, image quality and conditional diversity. We evaluate on I2I translation tasks on Edges→Handbags [16] scaled to 64×64 pixels and DIODE-Outdoor scaled to 256×256 [37], and Deblurring on ImageNet dataset [9]. For evaluation metrics, we use Fréchet Inception Distance (FID) [13] for all experiments, and additionally measure Inception Scores (IS) [3], Learned Perceptual Image Patch Similarity (LPIPS) [41], Mean Square Error (MSE), following previous works [42, 18]. In addition, we use VS and AFD, Eq. 14, to measure conditional diversity. Further details of the experiments and design guidelines are provided in Appendix G and E.

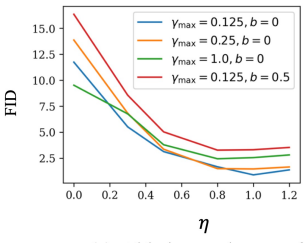
Sampler. We evaluate different sampling algorithms in Fig. 4 (a), the results demonstrate that setting $\epsilon_t = 0$ and using Eq. (12) for the last 2 steps can significantly improve sampled image



(a). Ablation study on discretization.



(b). Ablation study on γ_{max} for SDB model with Linear path.



(c). Ablation study on ϵ_t for SDB model with Linear path.

Figure 4: Ablation studies on discretization, γ_{max} and ϵ_t. (a). We evaluate different discretization schemes on Edges2handbags (64×64) dataset using DDBM-VP pretrained model, A represents simple Euler discretization in Eq. (13), B reprents setting $\epsilon_t = 0$ for the last 2 steps, C represents using Eq. (12) for $\epsilon_t = 0$. (b). Ablation study on γ_{max} evaluated by DIODE (64×64) dataset. (c). Ablation study on ϵ_t through our ECSI model with Linear path on Edges2handbags (64×64) dataset, where $\epsilon_t = \eta(\gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2)$.

Table 3: Quantitative results in the I2I translation task Edges2handbags (64×64) and DIODE (256×256) datasets. Our results were achieved by Linear transition kernel and setting $\eta = 1$.

		Edges→handbags (64×64)				DIODE-Outdoor (256×256)			
Model	**NFE**	**FID** ↓	**IS** ↑	**LPIPS** ↓	**MSE**	**FID** ↓	**IS** ↑	**LPIPS** ↓	**MSE**
Pix2Pix [16]	1	74.8	3.24	0.356	0.209	82.4	4.22	0.556	0.133
DDIB [36]	$\geq 40^\dagger$	186.84	2.04	0.869	1.05	242.3	4.22	0.798	0.794
SDEdit [25]	≥ 40	26.5	3.58	0.271	0.510	31.14	5.70	0.714	0.534
Rectified Flow [22]	≥ 40	25.3	2.80	0.241	0.088	77.18	5.87	0.534	0.157
I^2SB [21]	≥ 40	7.43	3.40	0.244	0.191	9.34	5.77	0.373	0.145
DDBM [18]	118	1.83	3.73	0.142	0.040	4.43	6.21	0.244	0.084
DBIM [42]	20	1.74	3.64	0.095	0.005	4.99	6.10	0.201	0.017
ECSI ($\gamma_{\text{max}} = 0.125$)	5	**0.89**	4.10	0.049	0.024	12.97	5.49	0.269	0.074
	10	**0.67**	4.11	0.045	0.024	10.12	5.56	0.255	0.076
	20	**0.56**	4.11	0.044	0.024	8.62	5.62	0.248	0.078
ECSI ($\gamma_{\text{max}} = 0.25$)	5	1.46	**4.21**	**0.040**	0.016	**4.16**	5.83	**0.104**	0.029
	10	1.38	**4.22**	**0.038**	0.017	**3.44**	5.86	**0.098**	0.029
	20	1.40	**4.20**	**0.038**	0.017	**3.27**	5.85	**0.094**	0.029

quality compared with simple Euler discretization and DDBM sampler. Furtheremore, By specifically designing noise control during sampling, our sampler surpasses the sampling results by DDBM and DBIM with the same pretrained model. The results are demonstrated in Table 2. We set the number of function evaluations (NFEs) from the set $[5, 10, 20]$.

Bridge paths. We introduced an extensive bridge design space and begin by focusing on linear transition paths with different strength of maximum stochasticity, i.e., $p_{t|0,T}(x_t|x_0, x_T) = \mathcal{N}(x_t; (1-t)x_0 + tx_T, \frac{1}{4}\gamma_{\text{max}}^2 t(1-t)\mathbb{I})$. We conducted detailed ablation studies on γ_{max} and η for the Linear path on DIODE (64×64) dataset, as shown in Fig. 4 (b) and (c). The optimal values for γ_{max} were found to be 0.125 and 0.25, while the best performance for η was achieved with $\eta = 0.8$ and $\eta = 1.0$. Performance deteriorates when either parameter is too small or too large. Based on the results of these ablation studies, we further trained ECSI models on the Edges2handbags (64×64) and DIODE (256×256) datasets by taking $\gamma_{\text{max}} \in \{0.125, 0.5\}$ and setting $\eta = 1.0$. The results are presented in Table 3. Our models establish a new benchmark for image quality, as evaluated by FID, IS and LPIPS. Despite our models having slightly higher MSEs compared to the baseline DDBM and DBIM, we believe that a larger MSE indicates that the generated images are distinct from their references, suggesting a richer diversity.

Modifying base distribution. Through controlling noise in the base distribution, we achieved a more diverse set of sample images, while this diversity comes at the cost of slightly higher FID scores and slower sampling speed. We show generated images in Fig. 5. More visualization can be found in Appendix I, which shows that by introducing booting noise to the input data distribution, the model can generate samples with more diverse colors and textures. Further quantitative results are presented

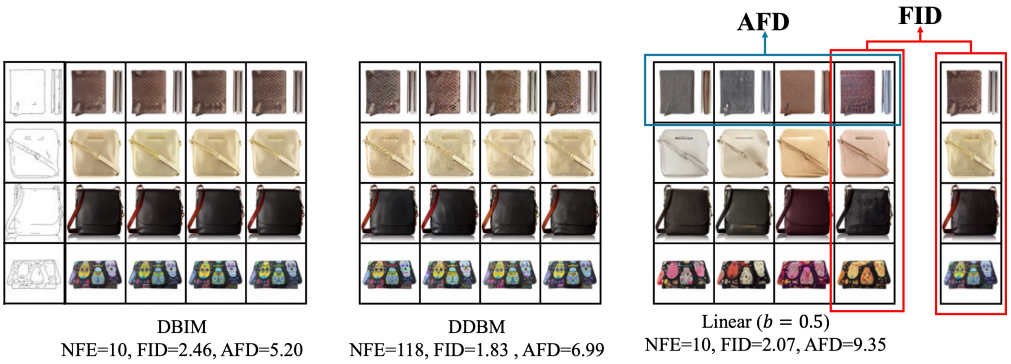


Figure 5: Visualization of conditional diversity via sampled images in a one-to-many translation task. While FID measures diversity within columns, AFD evaluates diversity across rows. The visualization further proved the effectiveness of AFD. More sampled images can be found in Appendix I.

Table 4: Quantitative results for Different denoisers and samplers on Edges2handbags (64×64). Our baseline is achieved by DDBM pretrained checkpoint and DBIM sampler.

Method	FID ↓			AFD ↑			VS ↑		
	NFE=5	NFE=10	NFE=20	NFE=5	NFE=10	NFE=20	NFE=5	NFE=10	NFE=20
DDBM (pre) + DBIM sampler	3.60	2.46	1.74	5.63	5.20	5.84	1.16	1.23	1.26
A: DDBM (pre) + ECSI sampler	2.36	2.25	1.53	5.11	5.70	6.04	1.15	1.20	1.23
B: ECSI (pre) + ECSI sampler	**0.89**	**0.67**	**0.56**	6.00	6.05	6.25	1.22	1.25	1.28
B + Modified base density	3.31	2.07	1.74	**8.53**	**9.35**	**9.65**	**1.48**	**1.63**	**1.69**

in Table 4, confirming that our model surpasses the vanilla DDBM in terms of image quality, sample efficiency, and conditional diversity.

Deblurring on ImageNet Dataset. We evaluate our models for Gaussian deblurring applying a Gaussian kernel with $\sigma = 10$ and Uniform deblurring, shown in Table 5. The results demonstrates that our ECSI models achieve much lower FID score.

7 Related Work

Diffusion Bridge Models. Diffusion bridges are faster diffusion processes that could learn the mapping between two random target distributions [39, 35], demonstrating significant potential in various areas, such as protein docking [32], mean-field game [20], I2I translation [21, 18]. According to different design philosophies, DBMs can be divided into two groups: bridge matching and stochastic interpolants. The idea of bridge matching was first proposed by Peluchetti et al. [28], and can be viewed as a generalization of score matching [34]. Based on this, diffusion Schrödinger bridge matching (DSBM) has been developed for solving Schrödinger bridge problems [35, 39]. In addition, Liu et al. [21] utilize bridge matching to perform image restoration tasks and noted benefits of noise empirically, the experiments shows the new model is more efficient and interpretable than score-based generative models [21]. Furthermore, our benchmark DDBM [18] achieve significant improvement for various I2I translation tasks, DBIM [42] improved the sampling algorithm for DDBM, significantly reducing sampling time while maintaining the same image quality.

Image-to-Image Translations. While diffusion models are strong at generating images, applying them to image-to-image (I2I) translation is more difficult due to artifacts in the output. DiffI2I improves quality and alignment with fewer diffusion steps [5]. In latent space, S2ST speeds up translation and reduces memory use [27]. Other methods improve guidance using features like frequency control [26, 15, 38]. A common challenge is that many models require joint training on both source and target domains, raising privacy concerns. Injecting-Diffusion tackles this by isolating shared content for unpaired translation [24]. SDDM improves interpretability by breaking down the score function across diffusion steps [30].

Table 5: Deblurring results with respect to different kernels, evaluated by FID on the 10k ImageNet (256×256) validation subset. Our results are achieved by 20 NFEs.

Kernel	DDRM	DDNM	Pallette	CDSB	I^2SB	ECSI (ours)
Uniform	9.9	3.0	4.1	15.5	3.9	**1.11**
Gaussian	6.1	2.9	3.1	7.7	3.0	**0.41**

8 Conclusion

We introduced Endpoint-Conditioned Stochastic Interpolants (ECSI)—an improved version of stochastic interpolants that adds endpoint conditioning, modifies the base distribution, and uses discretization to explore the design space of Diffusion Bridge Models (DBMs). We highlighted a key issue often overlooked: one-to-many image translation tasks lack conditional diversity. Our findings show that resolving this requires adjusting the starting distribution, not the path or sampler. ECSI sets new benchmarks in image quality, sampling efficiency, and conditional diversity on tasks like 64×64 edges2handbags, 256×256 DIODE-outdoor, and ImageNet deblurring.

Limitations. (i) We note that optimal path design may vary by task, leaving room for future refinement. (ii) Incorporating guidance techniques may further enhance model performance.

References

[1] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[2] Michael S Albergo, Mark Goldstein, Nicholas M Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic interpolants with data-dependent couplings. *arXiv preprint arXiv:2310.03725*, 2023.

[3] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

[4] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

[5] Bin Xia, Yulun Zhang, Shiyin Wang, Yitong Wang, Xiaohong Wu, Yapeng Tian, Wenge Yang, Radu Timotfe, and Luc Van Gool. DiffI2I: Efficient Diffusion Model for Image-to-Image Translation. *arXiv.org*, 2023.

[6] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-VAE. *arXiv preprint arXiv:1804.03599*, 2018.

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

[8] Valentin De Bortoli, Guan-Horng Liu, Tianrong Chen, Evangelos A Theodorou, and Weilie Nie. Augmented bridge matching. *arXiv preprint arXiv:2311.06978*, 2023.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[11] Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022.

[12] Nikita Gushchin, David Li, Daniil Selikhanovych, Evgeny Burnaev, Dmitry Baranchuk, and Alexander Korotin. Inverse bridge matching distillation. *arXiv preprint arXiv:2502.01362*, 2025.

[13] 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. *Advances in neural information processing systems*, 30, 2017.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[15] Hyunsoo Lee, Minsoo Kang, and Bohyung Han. Conditional Score Guidance for Text-Driven Image-to-Image Translation. *Neural Information Processing Systems*, 2023.

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[17] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

[18] Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising Diffusion Bridge Models. *arXiv.org*, 2023.

[19] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[20] Guan-Horng Liu, Tianrong Chen, Oswin So, and Evangelos Theodorou. Deep generalized schrödinger bridge. *Advances in Neural Information Processing Systems*, 35:9374–9388, 2022.

[21] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I^2sb: Image-to-image schrödinger bridge. *arXiv preprint arXiv:2302.05872*, 2023.

[22] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[23] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024.

[24] Luying Li and Lizhuang Ma. Injecting-Diffusion: Inject Domain-Independent Contents into Diffusion Models for Unpaired Image-to-Image Translation. *IEEE International Conference on Multimedia and Expo*, 2023.

[25] 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. *arXiv preprint arXiv:2108.01073*, 2021.

[26] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation. *Computer Vision and Pattern Recognition*, 2023.

[27] Or Greenberg, Eran Kishon, and Dani Lischinski. S2ST: Image-to-Image Translation in the Seed Space of Latent Diffusion. *arXiv.org*, 2023.

[28] Stefano Peluchetti. Non-denoising forward-time diffusions. *arXiv preprint arXiv:2312.14589*, 2023.

[29] L Chris G Rogers and David Williams. *Diffusions, Markov processes, and martingales: Itô calculus*, volume 2. Cambridge university press, 2000.

[30] Shurong Sun, Longhui Wei, Junliang Xing, Jia Jia, and Qi Tian. SDDM: Score-Decomposed Diffusion Models on Manifolds for Unpaired Image-to-Image Translation. *International Conference on Machine Learning*, 2023.

[31] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

[32] Vignesh Ram Somnath, Matteo Pariset, Ya-Ping Hsieh, Maria Rodriguez Martinez, Andreas Krause, and Charlotte Bunne. Aligned diffusion schrödinger bridges. In *Uncertainty in Artificial Intelligence*, pages 1985–1995. PMLR, 2023.

[33] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[34] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[35] Stefano Peluchetti. Diffusion Bridge Mixture Transports, Schr\"odinger Bridge Problems and Generative Modeling. *arXiv.org*, 2023.

[36] Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. *arXiv preprint arXiv:2203.08382*, 2022.

[37] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019.

[38] Xiang Gao, Zhengbo Xu, Junhan Zhao, and Jiaying Liu. Frequency-Controlled Diffusion Model for Versatile Text-Guided Image-to-Image Translation. *AAAI Conference on Artificial Intelligence*, 2024.

[39] Yifeng Shi, Valentin De Bortoli, Andrew T. Campbell, and Arnaud Doucet. Diffusion Schr\"odinger Bridge Matching. *Neural Information Processing Systems*, 2023.

[40] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pages 12310–12320. PMLR, 2021.

[41] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[42] Kaiwen Zheng, Guande He, Jianfei Chen, Fan Bao, and Jun Zhu. Diffusion bridge implicit models. *arXiv preprint arXiv:2405.15885*, 2024.

Table 6: Evaluation for generative models: ImageNet-1-mode, ImageNet-2-modes, ImageNet-5-modes, and ImageNet-10-modes.

Model	ImageNet-1-mode	ImageNet-2-modes	ImageNet-5-modes	ImageNet-10-modes
FID	58.30	57.34	57.78	57.26
AFD	0	8.14	12.84	14.47

A AFD validation

In this section, we thoroughly validate the effectiveness of our proposed metric, AFD, for measuring conditional diversity and demonstrate its role as a complementary metric to FID. In unconditional generation scenarios, the FID is widely used to evaluate the diversity of generated images. While low FID scores generally indicate high diversity across the entire dataset, they do not necessarily imply high conditional diversity. For instance, we observed that samples generated by the DDBM model often lack diversity when conditioned on edge images, despite achieving very low FID scores. To address this limitation, we introduce the concept of conditional diversity and propose a corresponding metric to quantify it.

The first question is why FID failed to measure the conditional diversity. To illustrate the limitations of FID in capturing conditional diversity, consider an extreme case: if the images generated by a generative model are identical to a set of baseline images, the FID score can be very low since the two distributions are indistinguishable. However, this scenario does not reflect diversity within the conditional outputs.

To further support our point, we designed two classes of pseudo-generative models capable of controlling the diversity of the generated images, which are further validated by FID and AFD. The experiments are evaluated on Imagenet dataset [9].

A.1 Pseudo-generative models by random selection

We designed four pseudo-generative models: ImageNet-1-mode, ImageNet-2-modes, ImageNet-5-modes, and ImageNet-10-modes. The experimental setup is as follows:

- We selected 11,000 samples from the ImageNet validation dataset, randomly choosing 11 images per class.
- From these, we designated 1,000 images as the "real" set, while the remaining images served as the source pool for the generative models.
- Each ImageNet-k-modes model simulates a generative process by randomly sampling images from a pool of k distinct images within a given class.

We present sampled images in Fig. 6, where it is evident that the ImageNet-10-modes model generates images with the highest conditional diversity. To quantify this, we conducted experiments to calculate both FID and AFD for the four generative models. The results are summarized in Table 6. While the FID scores are nearly identical across all models, the AFD values increase as the conditional diversity of the generative models improves. This highlights that AFD is a more effective metric for capturing conditional diversity than FID.

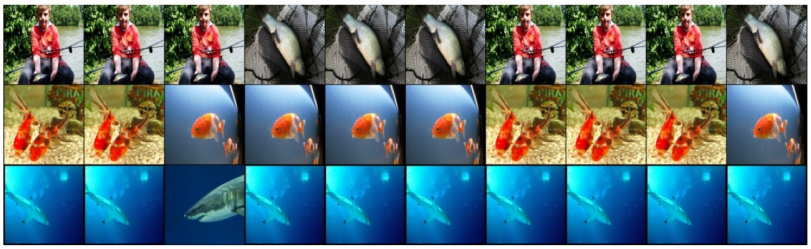
A.2 Pseudo-generative models by strong augmentation

Strong augmentation has been widely used in computer vision to generate synthetic data while preserving its underlying semantics [7, 40, 31, 4]. The intensity of augmentation can be adjusted, with higher intensities producing more diverse images. To further validate our proposed metric, AFD, as a measure of diversity, we construct pseudo-generative models using strong augmentation.

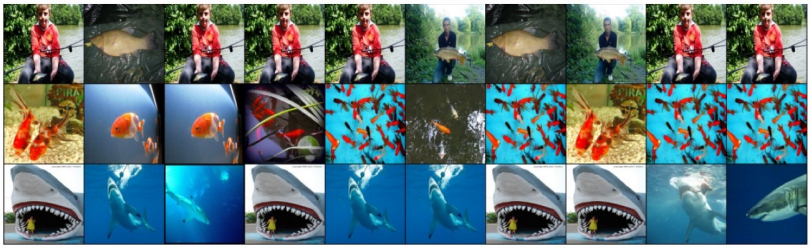
We selected 1,000 images from the ImageNet-1k dataset, one from each category. These images were subjected to data augmentation, specifically using ColorJitter, with varying magnitudes to enhance diversity. For each image, the augmentation was applied 16 times, creating an augmented dataset for

ImageNet-1-mode: FID=58.30, AFD=0

ImageNet-2-modes: FID=57.34, AFD=8.14



ImageNet-5-modes: FID=57.78, AFD=12.84



ImageNet-10-modes: FID=57.26, AFD=14.47

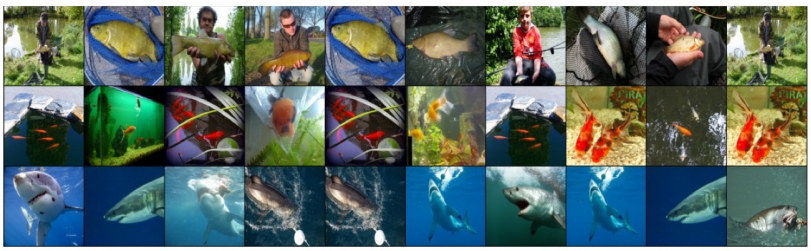


Figure 6: Sampled images from 4 generative models: ImageNet-1-mode, ImageNet-2-modes, ImageNet-5-modes, ImageNet-10-modes.

each magnitude setting. We then calculated the AFD for these augmented datasets to evaluate the relationship between dataset diversity (as influenced by augmentation magnitude) and the AFD value.

Table 7 summarizes the AFD results across various augmentation magnitude settings. The results show that as diversity increases, AFD values also rise, further confirming that the proposed AFD metric is a reliable indicator of image diversity.

B Relation to Stochastic Interpolants

Conditioned Stochastic Interpolants build a marginal probability path $p_{t|y}$ using a mixture of interpolating densities: $p_{t|y}(x) = \int p_t(x_t \mid x_0, x_1)\pi(x_0, x_1 \mid y)dx_0dx_1$, where $\pi(x_0, x_1 \mid y)$ is a joint distribution with marginals $\pi_{0|y}(x_0 \mid y)$ and $\pi_{1|y}(x_1 \mid y)$. For linear interpolants given by: $X_t = \alpha_t X_0 + \beta_t X_1 + \gamma_t z$. The conditional kernel $p_t(x_t \mid x_0, x_1)$ is given by a Gaussian distribution:

Table 7: AFD results across different augmentation magnitudes

Augmentation magnitude	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
AFD	2.16	3.77	5.13	6.16	6.98	7.63	8.22	9.01
FID	0.20	2.95	7.02	11.62	16.33	20.84	25.12	28.89

$p_t(x_t \mid x_0, x_1) = \mathcal{N}(\alpha_t x_0 + \beta_t x_1, \sigma_t^2 I), \forall\, t \in [0,1]$. Then we can sample from the conditional distribution $p_{0|y}(x_0 \mid y)$ by running a stochastic process $p_{t|y}(x_t \mid y)$ from time $t = 1$ to $t = 0$, which is given by the following SDE:

$$dx = b(t, x, y)dt + \sqrt{2\epsilon_t} dW_t, \quad x_1 \sim p_{1|y}, \tag{15}$$

where the drift term $b(t, x, y)$ is:

$$b(t, x, y) = \dot{\alpha}_t \mathbb{E}[x_0 \mid x, y] + \dot{\beta}_t \mathbb{E}[x_1 \mid x, y] + (\dot{\gamma}_t + \frac{\epsilon_t}{\gamma_t})\mathbb{E}[z \mid x, y] \tag{16}$$

As y represents null conditioning, Eq. (15) recover the original sampler of Stochastic Interpolants. In the drift term, $\mathbb{E}[x_0 \mid x, y]$, $\mathbb{E}[x_1 \mid x, y]$ and $\mathbb{E}[z \mid x, y]$ are unknown, but we only need to estimate two of them, since

$$\mathbb{E}[x_t \mid x_t, y] = \alpha_t \mathbb{E}[x_0 \mid x_t, y] + \beta_t \mathbb{E}[x_1 \mid x_t, y] + \gamma_t \mathbb{E}[z \mid x_t, y] = x_t$$

We can further reduce the number of unknown term by endpoint-conditioning. Here as we replace condition y to be endpoint x_1, the term $\mathbb{E}[x_1 \mid x, x_1] = x_1$. So we have:

$$b(t, x, y) = \dot{\alpha}_t \mathbb{E}[x_0 \mid x, x_1] + \dot{\beta}_t x_1 + (\dot{\gamma}_t + \frac{\epsilon_t}{\gamma_t})\mathbb{E}[z \mid x, x_1] \tag{17}$$

This is exactly the sampler for ECSI in Prop. 4.1. Therefore, both SI and ECSI can be seen as special cases of Conditioned SI. A key advantage of ECSI is its efficiency: while SI need to estimate two terms: $\mathbb{E}[x_0 \mid x_t]$ and $\mathbb{E}[x_1 \mid x_t]$, ECSI only estimates $\mathbb{E}[x_0 \mid x_t, y]$.

C Proofs

There are infinitely many pinned processes characterized by the Gaussian transition kernel $p_{t|0,T}(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T) = \mathcal{N}(\mathbf{x}_t; \alpha_t \mathbf{x}_0 + \beta_t \mathbf{x}_T, \gamma_t^2 \mathbb{I})$. Specifically, we formalize the pinned process as a linear Itô SDE, as presented in Lemma C.1.

Lemma C.1. *There exist a linear Itô SDE*

$$d\mathbf{X}_t = [f_t \mathbf{X}_t + s_t \mathbf{x}_T]dt + g_t d\mathbf{W}_t, \quad \mathbf{X}_0 = \mathbf{x}_0, \tag{18}$$

where $f_t = \frac{\dot{\alpha}_t}{\alpha_t}$, $s_t = \dot{\beta}_t - \frac{\dot{\alpha}_t}{\alpha_t}\beta_t$, $g_t = \sqrt{2(\gamma_t \dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2)}$, *that has a Gaussian marginal distribution* $\mathcal{N}\left(\mathbf{x}_t; \alpha_t \mathbf{x}_0 + \beta_t \mathbf{x}_T, \gamma_t^2 \mathbb{I}\right)$.

Proof. Let $\mathbf{m}_t$ denote the mean function of the given Itô SDE, then we have $\frac{d\mathbf{m}_t}{dt} = f_t \mathbf{m}_t + s_t \mathbf{x}_T$. Given the transition kernel, the mean function $\mathbf{m}_t = \alpha_t \mathbf{x}_0 + \beta_t \mathbf{x}_T$, therefore,

$$\dot{\alpha}_t \mathbf{x}_0 + \dot{\beta}_t \mathbf{x}_T = f_t(\alpha_t \mathbf{x}_0 + \beta_t \mathbf{x}_T) + s_t \mathbf{x}_T. \tag{19}$$

Matching the above equation:

$$f_t = \frac{\dot{\alpha}_t}{\alpha_t}, s_t = \dot{\beta}_t - \beta_t \frac{\dot{\alpha}_t}{\alpha_t}. \tag{20}$$

Further, For the variance γ_t^2 of the process, the dynamics are given by:

$$\frac{d\gamma_t^2}{dt} = 2f_t\gamma_t^2 + g_t^2. \tag{21}$$

Solving for g_t^2, we substitute $f_t = \frac{\dot{\alpha}_t}{\alpha_t}$:

$$g_t^2 = \frac{d\gamma_t^2}{dt} - 2\frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2 \tag{22}$$

Therefore,

$$g_t = \sqrt{2(\gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2)}. \tag{23}$$

□

Given the pinned process (18), we can sample from the conditional distribution $p_{0|T}(\mathbf{x}_0|\mathbf{x}_T)$ by solving the reverse SDE or ODE from $t = T$ to $t = 0$:

$$d\mathbf{X}_t = \left[f_t\mathbf{X}_t + s_t\mathbf{x}_T - g_t^2\nabla_{\mathbf{X}_t}\log p_t(\mathbf{X}_t|\mathbf{x}_T)\right]dt + g_t d\mathbf{W}_t, \quad \mathbf{X}_T = \mathbf{x}_T, \tag{24}$$

$$d\mathbf{X}_t = \left[f_t\mathbf{X}_t + s_t\mathbf{x}_T - \frac{1}{2}g_t^2\nabla_{\mathbf{X}_t}\log p_t(\mathbf{X}_t|\mathbf{x}_T)\right]dt \quad \mathbf{X}_T = \mathbf{x}_T, \tag{25}$$

where the score $\nabla_{\mathbf{X}_t}\log p_t(\mathbf{X}_t|\mathbf{x}_T)$ can be estimated by score matching objective (5).

For dynamics described by ODE $d\mathbf{X}_t = \mathbf{u}_t dt$, we can identify the entire class of SDEs that maintain the same marginal distributions, as detailed in Lemma C.2. This enables us to control the noise during sampling by appropriately designing ϵ_t.

Lemma C.2. *Consider a continuous dynamics given by ODE of the form: $d\mathbf{X}_t = \mathbf{u}_t dt$, with the density evolution $p_t(\mathbf{X}_t)$. Then there exists forward SDEs and backward SDEs that match the marginal distribution p_t. The forward SDEs are given by: $d\mathbf{X}_t = (\mathbf{u}_t + \epsilon_t\nabla\log p_t)dt + \sqrt{2\epsilon_t}d\mathbf{W}_t, \epsilon_t > 0$. The backward SDEs are given by: $d\mathbf{X}_t = (\mathbf{u}_t - \epsilon_t\nabla\log p_t)dt + \sqrt{2\epsilon_t}d\mathbf{W}_t, \epsilon_t > 0$.*

Proof. For the forward SDEs, the Fokker-Planck equations are given by:

$$\frac{\partial p_t(\mathbf{X}_t)}{\partial t} = -\nabla\cdot\left[\left(\mathbf{u}_t + \epsilon_t\nabla\log p_t\right)p_t(\mathbf{X}_t)\right] + \epsilon_t\nabla^2 p_t(\mathbf{X}_t) \tag{26}$$

$$= -\nabla\cdot\left[\mathbf{u}_t p_t(\mathbf{X}_t)\right] - \nabla\cdot\left[\epsilon_t(\nabla\log p_t)p_t(\mathbf{X}_t)\right] + \epsilon_t\nabla^2 p_t(\mathbf{X}_t) \tag{27}$$

$$= -\nabla\cdot\left[\mathbf{u}_t p_t(\mathbf{X}_t)\right] - \epsilon_t\nabla\cdot\left[\nabla p_t(\mathbf{X}_t)\right] + \epsilon_t\nabla^2 p_t(\mathbf{X}_t) \tag{28}$$

$$= -\nabla\cdot\left[\mathbf{u}_t p_t(\mathbf{X}_t)\right]. \tag{29}$$

This is exactly the Fokker-Planck equation for the original deterministic ODE $d\mathbf{X}_t = \mathbf{u}_t\, dt$. Therefore, the forward SDE maintains the same marginal distribution $p_t(\mathbf{X}_t)$ as the original ODE.

Now consider the backward SDEs, the Fokker-Planck equations become:

$$\frac{\partial p_t(\mathbf{X}_t)}{\partial t} = -\nabla\cdot\left[\left(\mathbf{u}_t - \epsilon_t\nabla\log p_t\right)p_t(\mathbf{X}_t)\right] - \epsilon_t\nabla^2 p_t(\mathbf{X}_t) \tag{30}$$

$$= -\nabla\cdot\left[\mathbf{u}_t p_t(\mathbf{X}_t)\right] + \nabla\cdot\left[\epsilon_t(\nabla\log p_t)p_t(\mathbf{X}_t)\right] - \epsilon_t\nabla^2 p_t(\mathbf{X}_t) \tag{31}$$

$$= -\nabla\cdot\left[\mathbf{u}_t p_t(\mathbf{X}_t)\right]. \tag{32}$$

This is again the Fokker-Planck equation corresponding to the original deterministic ODE $d\mathbf{X}_t = \mathbf{u}_t\, dt$. Therefore, the backward SDE also maintains the same marginal distribution $p_t(\mathbf{X}_t)$.

□

Lemma C.3. *Let $(\mathbf{x}_0, \mathbf{x}_T) \sim \pi_0(\mathbf{x}_0, \mathbf{x}_T)$, $\mathbf{x}_t \sim p_t(\mathbf{x}|\mathbf{x}_0, \mathbf{x}_T)$, Given the transition kernel: $p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T) = \mathcal{N}\left(\mathbf{x}_t; \alpha_t \mathbf{x}_0 + \beta_t \mathbf{x}_T, \gamma_t^2 \mathbb{I}\right)$, if $\hat{\mathbf{x}}_0(\mathbf{x}_t, \mathbf{x}_T, t)$ is a denoiser function that minimizes the expected L_2 denoising error for samples drawn from $\pi_0(\mathbf{x}_0, \mathbf{x}_T)$:*

$$\hat{\mathbf{x}}_0(\mathbf{x}_t, \mathbf{x}_T, t) = \arg \min_{D(\mathbf{x}_t, \mathbf{x}_T, t)} \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_T, \mathbf{x}_t} \left[\lambda(t) \|D(\mathbf{x}_t, \mathbf{x}_T, t) - \mathbf{x}_0\|_2^2\right], \tag{33}$$

then the score has the following relationship with $\hat{\mathbf{x}}_0(\mathbf{x}_t, \mathbf{x}_T, t)$:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_T) = \frac{\alpha_t \hat{\mathbf{x}}_0(\mathbf{x}_t, \mathbf{x}_T, t) + \beta_t \mathbf{x}_T - \mathbf{x}_t}{\gamma_t^2}. \tag{34}$$

Proof.

$$\mathcal{L}(D) = \mathbb{E}_{(\mathbf{x}_0, \mathbf{x}_T) \sim \pi_0(\mathbf{x}_0, \mathbf{x}_T)} \mathbb{E}_{\mathbf{x}_t \sim p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_T)} \|D(\mathbf{x}_t) - \mathbf{x}_0\|_2^2 \tag{35}$$

$$= \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \underbrace{\int_{\mathbb{R}^d} p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_T) \pi_0(\mathbf{x}_0, \mathbf{x}_T) \|D(\mathbf{x}_t) - \mathbf{x}_0\|_2^2 \, \mathrm{d}\mathbf{x}_0}_{=: \mathcal{L}(D; \mathbf{x}_t, \mathbf{x}_T)} \, \mathrm{d}\mathbf{x}_T \mathrm{d}\mathbf{x}_t, \tag{36}$$

$$\mathcal{L}(D; \mathbf{x}_t, \mathbf{x}_T) = \int_{\mathbb{R}^d} p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_T) \pi_0(\mathbf{x}_0, \mathbf{x}_T) \|D(\mathbf{x}_t) - \mathbf{x}_0\|_2^2 \, \mathrm{d}\mathbf{x}_0, \tag{37}$$

we can minimize $\mathcal{L}(D)$ by minimizing $\mathcal{L}(D; \mathbf{x}_t, \mathbf{x}_T)$independently for each $\{\mathbf{x}_t, \mathbf{x}_T\}$ pair.

$$D^*(\mathbf{x}_t, \mathbf{x}_T) = \arg \min_{D(\mathbf{x}_t)} \mathcal{L}(D; \mathbf{x}_t, \mathbf{x}_T) \tag{38}$$

$$\mathbf{0} = \nabla_{D(\mathbf{x}_t, \mathbf{x}_T)} [\mathcal{L}(D; \mathbf{x}_t, \mathbf{x}_T)] \tag{39}$$

$$= \int_{\mathbb{R}^d} p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_T) \pi_0(\mathbf{x}_0, \mathbf{x}_T) 2[D(\mathbf{x}, \mathbf{x}_T) - \mathbf{x}_0] \, \mathrm{d}\mathbf{x}_0 \tag{40}$$

$$= 2[D(\mathbf{x}_t, \mathbf{x}_T) \int_{\mathbb{R}^d} p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_T) \pi_0(\mathbf{x}_0, \mathbf{x}_T) \, \mathrm{d}\mathbf{x}_0 - \int_{\mathbb{R}^d} p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_T) \pi_0(\mathbf{x}_0, \mathbf{x}_T) \mathbf{x}_0 \, \mathrm{d}\mathbf{x}_0] \tag{41}$$

$$= 2[D(\mathbf{x}) p_t(\mathbf{x}_t, \mathbf{x}_T) - \int_{\mathbb{R}^d} p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_T) \pi_0(\mathbf{x}_0, \mathbf{x}_T) \mathbf{x}_0 \, \mathrm{d}\mathbf{x}_0], \tag{42}$$

$$D^*(\mathbf{x}_t, \mathbf{x}_T) = \int_{\mathbb{R}^d} \frac{p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_T) \pi_0(\mathbf{x}_0, \mathbf{x}_T) \mathbf{x}_0}{p_t(\mathbf{x}_t, \mathbf{x}_T)} \, \mathrm{d}\mathbf{x}_0, \tag{43}$$

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_T) = \frac{\nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t, \mathbf{x}_T)}{p_t(\mathbf{x}_t, \mathbf{x}_T)} \tag{44}$$

$$= \frac{\int \nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t|\mathbf{x}_T, \mathbf{x}_0) \pi_0(\mathbf{x}_0, \mathbf{x}_T) d\mathbf{x}_0}{p_t(\mathbf{x}_t, \mathbf{x}_T)} \tag{45}$$

$$= -\int \frac{\mathbf{x}_t - \alpha_t \mathbf{x}_0 - \beta_t \mathbf{x}_T}{\gamma^2} \frac{p_t(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_T) \pi_0(\mathbf{x}_0, \mathbf{x}_T)}{p_t(\mathbf{x}_t, \mathbf{x}_T)} d\mathbf{x}_0 \tag{46}$$

$$= \frac{\alpha_t D^*(\mathbf{x}_t, \mathbf{x}_T) + \beta_t \mathbf{x}_T - \mathbf{x}_t}{\gamma^2}. \tag{47}$$

Thus we conclude the proof.

□

Proof of Prop. 4.1.

Table 8: Specify design choices for different model families. In the implementation, $\sigma_t = t$ for EDM, $\sigma_t = t, a_t = 1$ for DDBM-VE, $\sigma_t = \sqrt{e^{\frac{1}{2}\beta_d t^2 + \beta_{\min} t} - 1}$ and $a_t = 1/\sqrt{e^{\frac{1}{2}\beta_d t^2 + \beta_{\min} t}}$ for DDBM-VP, where β_d and $\beta_{\min}$ are parameters. We include details and proofs in Appendix D.

		I2SB	DDBM	DBIM	EDM	Ours
Transition kernel	α_t	$1 - \sigma_t^2/\sigma_T^2$	$a_t(1-a_T^2\sigma_t^2/(\sigma_t^2 a_t^2))$	$a_t(1-a_T^2\sigma_t^2/(\sigma_t^2 a_t^2))$	1	$1-t$
	β_t	σ_t^2/σ_T^2	$a_T\sigma_t^2/(\sigma_t^2 a_t)$	$a_T\sigma_t^2/(\sigma_t^2 a_t)$	0	t
	γ_t^2	$\sigma_t^2(1-\sigma_t^2/\sigma_T^2)$	$\sigma_t^2(1-a_T^2\sigma_t^2/(\sigma_t^2 a_t^2))$	$\sigma_t^2(1-a_T^2\sigma_t^2/(\sigma_t^2 a_t^2))$	σ_t^2	$\frac{\gamma_{\max}^2}{4}t(1-t)$
Sampling SDEs	ϵ_t	$\frac{\gamma_{t-\Delta t}^2\beta_t^2 - \beta_{t-\Delta t}^2\gamma_t^2}{2\beta_t^2\Delta t}$	$\eta(\gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2)$ $\eta = 0$ or $\eta = 1$	$\begin{cases}\frac{\gamma_{t-\Delta t}^2}{2\Delta t}, & t = 0\\ 0, & t \neq 0\end{cases}$	$\bar{\beta}_t\sigma_t^2$ -	$\eta(\gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2)$ $\eta \in [0,1]$
Base distribution	π_T	π_{cond}	π_{cond}	π_{cond}	π_{cond}	$\pi_{\text{cond}} * \mathcal{N}(0, b^2\mathbb{I})$
Discretization	-	Euler Eq. (12)	Euler Eq. (13)	Euler Eq. (12)	Heun -	Euler Eqs. (13) and (12)

Proof. Recall Eqs. (24) (25) and Lemma C.2,

$$d\mathbf{X}_t = \left[\frac{\dot{\alpha}_t}{\alpha_t}\mathbf{x}_t + (\dot{\beta}_t - \frac{\dot{\alpha}_t}{\alpha_t}\beta_t)\mathbf{x}_T - (\gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2 + \epsilon_t)\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t|\mathbf{x}_T)\right]dt + \sqrt{2\epsilon_t}d\mathbf{w}_t. \quad (48)$$

□

Next we take the reparameterized score in Eq. (34) into Eq. (48):

$$d\mathbf{X}_t = \left[\frac{\dot{\alpha}_t}{\alpha_t}\mathbf{X}_t + (\dot{\beta}_t - \frac{\dot{\alpha}_t}{\alpha_t}\beta_t)\mathbf{x}_T - (\gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2 + \epsilon_t)\frac{\alpha_t\hat{\mathbf{x}}_0 + \beta_t\mathbf{x}_T - \mathbf{X}_t}{\gamma_t^2}\right]dt + \sqrt{2\epsilon_t}d\mathbf{w}_t \quad (49)$$

$$= \left[\dot{\alpha}_t\hat{\mathbf{x}}_0 + \dot{\beta}_t\mathbf{x}_T - (\gamma_t\dot{\gamma}_t + \epsilon_t)\frac{\alpha_t\hat{\mathbf{x}}_0 + \beta_t\mathbf{x}_T - \mathbf{X}_t}{\gamma_t^2}\right]dt + \sqrt{2\epsilon_t}d\mathbf{w}_t \quad (50)$$

$$= \left[\dot{\alpha}_t\hat{\mathbf{x}}_0 + \dot{\beta}_t\mathbf{x}_T - (\dot{\gamma}_t + \frac{\epsilon_t}{\gamma_t})\frac{\alpha_t\hat{\mathbf{x}}_0 + \beta_t\mathbf{x}_T - \mathbf{X}_t}{\gamma_t}\right]dt + \sqrt{2\epsilon_t}d\mathbf{w}_t \quad (51)$$

$$= \left[\dot{\alpha}_t\hat{\mathbf{x}}_0 + \dot{\beta}_t\mathbf{x}_T - (\dot{\gamma}_t + \frac{\epsilon_t}{\gamma_t})\hat{\mathbf{z}}\right]dt + \sqrt{2\epsilon_t}d\mathbf{w}_t. \quad (52)$$

D Reframing previous methods in our framework

We draw a link between our framework and the diffusion bridge models used in DDBM.

D.1 DDBM-VE

DDBM-VE can be reformulated in our framework as we set :

$$\alpha_t = s_t(1 - \frac{\sigma_t^2}{\sigma_T^2}), \beta_t = \frac{s_t\sigma_t^2}{s_1\sigma_T^2}, \gamma_t = \sigma_t s_t\sqrt{(1 - \frac{\sigma_t^2}{\sigma_T^2})} \quad (53)$$

Proof. In the origin DDBM paper, the evolution of conditional probability $q(\mathbf{x}_t|\mathbf{x}_T)$ has a time reversed SDE of the form:

$$d\mathbf{X}_t = \left[\bar{\mathbf{f}}_t(\mathbf{X}_t) - g_t^2\bar{\mathbf{h}}_t(\mathbf{X}_t) - g_t^2\mathbf{s}_t(\mathbf{X}_t)\right]dt + g_t d\hat{\mathbf{W}}_t, \quad (54)$$

and an associated probability flow ODE

$$d\mathbf{X}_t = \left[\bar{\mathbf{f}}_t(\mathbf{X}_t) - g_t^2\bar{\mathbf{h}}_t(\mathbf{X}_t) - \frac{1}{2}g_t^2\mathbf{s}_t(\mathbf{X}_t)\right]dt. \tag{55}$$

Compare Eqs. (54) and 55 with Lemma C.1. We only need to prove:

$$\bar{\mathbf{f}}_t(\mathbf{X}_t) - g_t^2\bar{\mathbf{h}}_t(\mathbf{X}_t) = f_t\mathbf{X}_t + s_t\mathbf{x}_T, g_t = g_t. \tag{56}$$

In the original paper,

$$\bar{\mathbf{f}}_t(\mathbf{X}_t) = 0, g_t^2 = \frac{d}{dt}\sigma_t^2, \bar{\mathbf{h}}_t(\mathbf{X}_t) = \frac{\mathbf{x}_T - \mathbf{x}_t}{\sigma_T^2 - \sigma_t^2}. \tag{57}$$

Therefore,

$$\bar{\mathbf{f}}_t(\mathbf{X}_t) - g_t^2\bar{\mathbf{h}}_t(\mathbf{X}_t) = \frac{2\sigma_t\dot{\sigma}_t(\mathbf{x}_T - \mathbf{x}_t)}{\sigma_T^2 - \sigma_t^2}, g_t^2 = 2\dot{\sigma}_t\sigma_t. \tag{58}$$

In our framework, f_t, s_t, g_t^2 can be calculated:

$$f_t = \frac{\dot{\alpha}_t}{\alpha_t} = \frac{d}{dt}\log\alpha_t = \frac{d}{dt}\log\frac{\sigma_T^2 - \sigma_t^2}{\sigma_T^2} = \frac{-2\sigma_t\dot{\sigma}_t}{\sigma_T^2 - \sigma_t^2}, \tag{59}$$

$$s_t = \dot{\beta}_t - \frac{\dot{\alpha}_t}{\alpha_t}\beta_t = \frac{2\sigma_t\dot{\sigma}_t}{\sigma_T^2} + \frac{2\sigma_t\dot{\sigma}_t}{\sigma_T^2 - \sigma_t^2}\cdot\frac{\sigma_t^2}{\sigma_T^2} = \frac{2\sigma_t\dot{\sigma}_t}{\sigma_T^2 - \sigma_t^2}. \tag{60}$$

$$g_t^2 = 2(\gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2) = 2\gamma_t^2\left(\frac{\dot{\gamma}_t}{\gamma_t} - \frac{\dot{\alpha}_t}{\alpha_t}\right) = \gamma_t^2\left(\frac{(\sigma_T^2 - 2\sigma_t^2)\dot{\sigma}_t}{(\sigma_T^2 - \sigma_t^2)\sigma_t} + \frac{2\dot{\sigma}_t\sigma_t}{\sigma_T^2 - \sigma_t^2}\right) = 2\sigma_t\dot{\sigma}_t. \tag{61}$$

Therefore,

$$f_t\mathbf{X}_t + s_t\mathbf{x}_T = \frac{2\sigma_t\dot{\sigma}_t(\mathbf{x}_T - \mathbf{x}_t)}{\sigma_T^2 - \sigma_t^2} = \bar{\mathbf{f}}_t(\mathbf{X}_t) - g_t^2\bar{\mathbf{h}}_t(\mathbf{X}_t), \quad g_t = g_t, \tag{62}$$

which matches the formulation in DDBM.

□

D.2 DDBM-VP

DDBM-VP can be reformulated in our framework as we set :

$$\alpha_t = a_t(1 - \frac{\sigma_t^2 a_1^2}{\sigma_1^2 a_t^2}), \beta_t = \frac{\sigma_t^2 a_1}{\sigma_1^2 a_t}, \gamma_t = \sqrt{\sigma_t^2(1 - \frac{\sigma_t^2 a_1^2}{\sigma_1^2 a_t^2})}. \tag{63}$$

Proof. In the original DDBM-VP setting,

$$\bar{\mathbf{f}}_t(\mathbf{X}_t) = \frac{d\log a_t}{dt}\mathbf{x}_t, \tag{64}$$

$$g_t^2 = 2\sigma_t\dot{\sigma}_t - 2\frac{\dot{a}_t}{a_t}\sigma_t^2 = \frac{2\sigma_t\dot{\sigma}_t a_t - 2\sigma_t^2\dot{a}_t}{a_t}, \tag{65}$$

$$\bar{\mathbf{h}}_t(\mathbf{X}_t) = \frac{(a_t/a_1)\mathbf{x}_T - \mathbf{x}_t}{\sigma_t^2(\mathrm{SNR}_t/\mathrm{SNR}_1 - 1)} = \frac{a_1 a_t\mathbf{x}_T - a_1^2\mathbf{x}_t}{\sigma_1^2 a_t^2 - \sigma_t^2 a_1^2}. \tag{66}$$

Therefore,

$$\bar{\mathbf{f}}_t(\mathbf{X}_t) - g_t^2 \bar{\mathbf{h}}_t(\mathbf{X}_t) = \left[\frac{\dot{a}_t}{a_t} - \frac{2\sigma_t a_1^2(\dot{\sigma}_t a_t - \sigma_t \dot{a}_t)}{a_t(\sigma_1^2 a_t^2 - \sigma_t^2 a_1^2)} \right] \mathbf{x}_t + \frac{2\sigma_t a_1(\dot{\sigma}_t a_t - \sigma_t \dot{a}_t)}{\sigma_1^2 a_t^2 - \sigma_t^2 a_1^2} \mathbf{x}_T. \tag{67}$$

In our framework, f_t, s_t, g_t^2 can be calculated:

$$\begin{aligned} f_t = \frac{\dot{\alpha}_t}{\alpha_t} &= \frac{d}{dt} \log \alpha_t && (68)\\ &= \frac{d}{dt} \log \frac{\sigma_1^2 a_t^2 - \sigma_t^2 a_1^2}{\sigma_1^2 a_t} && (69)\\ &= \frac{2\sigma_1^2 a_t \dot{a}_t - 2a_1^2 \sigma_t \dot{\sigma}_t}{\sigma_1^2 a_t^2 - \sigma_t^2 a_1^2} - \frac{\dot{a}_t}{a_t} && (70)\\ &= \frac{\dot{a}_t}{a_t} - \frac{2a_1^2 \sigma_t(a_t \dot{\sigma}_t - \dot{a}_t \sigma_t)}{a_t(\sigma_1^2 a_t^2 - \sigma_t^2 a_1^2)}, && (71) \end{aligned}$$

$$\begin{aligned} s_t = \dot{\beta}_t - \frac{\dot{\alpha}_t}{\alpha_t}\beta_t = \beta_t(\frac{\dot{\beta}_t}{\beta_t} - \frac{\dot{\alpha}_t}{\alpha_t}) && (72)\\ = \frac{\sigma_t^2 a_1}{\sigma_1^2 a_t} \left(\frac{2\dot{\sigma}_t}{\sigma_t} - \frac{2\sigma_1^2 a_t \dot{a}_t - 2a_1^2 \sigma_t \dot{\sigma}_t}{\sigma_1^2 a_t^2 - \sigma_t^2 a_1^2} \right) && (73)\\ = \frac{2\sigma_t a_1(\dot{\sigma}_t a_t - \sigma_t \dot{a}_t)}{\sigma_1^2 a_t^2 - \sigma_t^2 a_1^2}, && (74) \end{aligned}$$

$$\begin{aligned} g_t^2 = \gamma_t \dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2 = \gamma_t^2 \left(\frac{\dot{\gamma}_t}{\gamma_t} - \frac{\dot{\alpha}_t}{\alpha_t} \right) && (75)\\ = \gamma^2 \frac{d}{dt} \log \frac{\gamma_t}{\alpha_t} && (76)\\ = \gamma^2 \frac{d}{dt} (\frac{1}{2} \log \frac{\sigma_t^2 \sigma_1^2}{\sigma_1^2 a_t^2 - \sigma_t^2 a_1^2}) && (77)\\ = \sigma_t^2 \left(1 - \frac{\sigma_t^2 a_1^2}{\sigma_1^2 a_t^2} \right) \left(\frac{\dot{\sigma}_t}{\sigma_t} - \frac{\sigma_1^2 a_t \dot{a}_t - a_1^2 \sigma_t \dot{\sigma}_t}{\sigma_1^2 a_t^2 - \sigma_t^2 a_1^2} \right) && (78)\\ = \frac{\dot{\sigma}_t \sigma_t a_t - \sigma_t^2 \dot{a}_t}{a_t}. && (79) \end{aligned}$$

Therefore,

$$f_t \mathbf{X}_t + s_t \mathbf{x}_T == \bar{\mathbf{f}}_t(\mathbf{X}_t) - g_t^2 \bar{\mathbf{h}}_t(\mathbf{X}_t), g_t = g_t, \tag{80}$$

which matches the formulation in DDBM.

□

D.3 EDM

ODE formulation. The ODE formulation in EDM can be formlated in our framework as we set $\alpha_t = 1, \beta_t = 0, \gamma_t = \sigma_t$.

Proof. Recall 25, the ODE formulation is given by:

$$d\mathbf{X}_t = \left[f_t \mathbf{X}_t + s_t \mathbf{x}_T - \frac{1}{2} g_t^2 \nabla_{\mathbf{X}_t} \log p_t(\mathbf{X}_t | \mathbf{x}_T) \right] dt \quad \mathbf{X}_T = \mathbf{x}_T \tag{81}$$

where $f_t = \frac{\dot{\alpha}_t}{\alpha_t}, \quad s_t = \dot{\beta}_t - \frac{\dot{\alpha}_t}{\alpha_t}\beta_t, \quad g_t = \sqrt{2(\gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2)}$. As $\alpha_t = 1, \beta_t = 0, \gamma_t = \sigma_t$, The sampling ODE is given by:

$$d\mathbf{X}_t = -\sigma_t\dot{\sigma}_t\nabla_{\mathbf{x}_t}\log p_t(\mathbf{X}_t)dt \tag{82}$$

□

Sampling SDEs with noise added. Recall Proposition 4.1, as $\alpha_t = 1, \beta_t = 0, \gamma_t = \sigma_t$, then the SDE has the form:

$$d\mathbf{X}_t = (-\sigma_t\dot{\sigma}_t + \epsilon_t)\nabla_{\mathbf{x}_t}\log p_t(\mathbf{X}_t)dt + \sqrt{2\epsilon_t}d\mathbf{W}_t. \tag{83}$$

Now we recover the stochastic sampling SDE in original EDM paper.

D.4 I2SB

I2SB can be reformulated in our framework as we let:

$$\alpha_t = 1 - \frac{\sigma_t^2}{\sigma_1^2}, \beta_t = \frac{\sigma_t^2}{\sigma_1^2}, \gamma_t = \sqrt{\sigma_t^2(1 - \frac{\sigma_t^2}{\sigma_1^2})} \tag{84}$$

where $\sigma_t^2 := \int_0^t \beta_\tau d\tau$.

When $2\epsilon_t\Delta t = \gamma_{t-\Delta t}^2 - \beta_{t-\Delta t}^2\gamma_t^2/\beta_t^2$, the coefficient of x_T in Eq. (12) vanishes. Thus, Eq. (12) can be simplified as:

$$x_{t-\Delta t} = (\alpha_{t-\Delta t} - \alpha_t\frac{\beta_{t-\Delta t}}{\beta_t})\hat{x}_0 + \frac{\beta_{t-\Delta t}}{\beta_t}x_t + \sqrt{\gamma_{t-\Delta t}^2 - \frac{\beta_{t-\Delta t}^2\gamma_t^2}{\beta_t^2}}\bar{z}_t \tag{85}$$

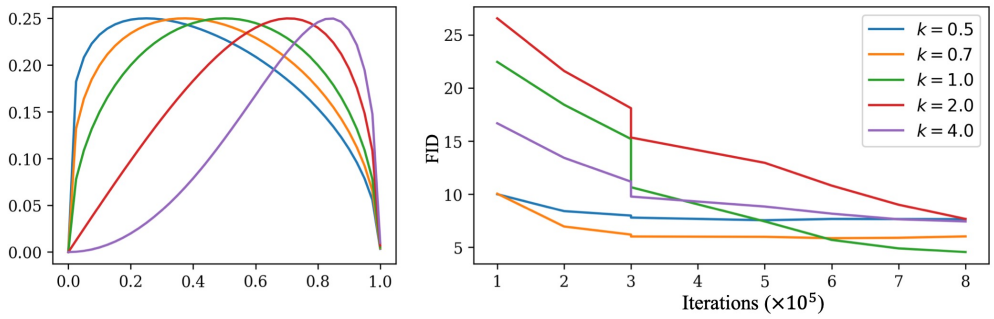
Using discretization in Eq. (85):

$$\begin{aligned}
\mathbf{x}_{t-\Delta t} &= (\alpha_{t-\Delta t} - \alpha_t\frac{\beta_{t-\Delta t}}{\beta_t})\hat{\mathbf{x}}_0 + \frac{\beta_{t-\Delta t}}{\beta_t}\mathbf{x}_t + \sqrt{\gamma_{t-\Delta t}^2 - \frac{\beta_{t-\Delta t}^2\gamma_t^2}{\beta_t^2}}\bar{\mathbf{z}}_t && (86)\\
&= (1 - \frac{\beta_{t-\Delta t}}{\beta_t})\hat{\mathbf{x}}_0 + \frac{\beta_{t-\Delta t}}{\beta_t}\mathbf{x}_t + \sqrt{\gamma_{t-\Delta t}^2 - \frac{\beta_{t-\Delta t}^2\gamma_t^2}{\beta_t^2}}\bar{\mathbf{z}}_t && (87)\\
&= (1 - \frac{\sigma_{t-\Delta t}^2}{\sigma_t^2})\hat{\mathbf{x}}_0 + \frac{\sigma_{t-\Delta t}^2}{\sigma_t^2}\mathbf{x}_t + \sqrt{\frac{\sigma_{t-\Delta t}^2(1 - \frac{\sigma_{t-\Delta t}^2}{\sigma_1^2})\frac{\sigma_t^4}{\sigma_1^4} - \frac{\sigma_{t-\Delta t}^4}{\sigma_1^4}\sigma_t^2(1 - \frac{\sigma_t^2}{\sigma_1^2})}{\frac{\sigma_t^4}{\sigma_1^4}}}\bar{\mathbf{z}}_t && (88)\\
&= (1 - \frac{\sigma_{t-\Delta t}^2}{\sigma_t^2})\hat{\mathbf{x}}_0 + \frac{\sigma_{t-\Delta t}^2}{\sigma_t^2}\mathbf{x}_t + \sqrt{\frac{\sigma_{t-\Delta t}^2(\sigma_t^2 - \sigma_{t-\Delta t}^2)}{\sigma_t^2}}\bar{\mathbf{z}}_t && (89)
\end{aligned}$$

In the I2SB paper, define $a_n^2 := \int_{t_n}^{t_{n+1}} \beta_\tau \mathrm{d}\tau$, $\sigma_n^2 := \int_0^{t_n} \beta_\tau \mathrm{d}\tau$. Therefore,

$$\mathbf{x}_n = \frac{a_n^2}{a_n^2 + \sigma_n^2}\hat{\mathbf{x}}_0 + \frac{\sigma_n^2}{a_n^2 + \sigma_n^2}\mathbf{x}_{n+1} + \sqrt{\frac{\sigma_n^2 a_n^2}{\alpha_n^2 + \sigma_n^2}}\bar{\mathbf{z}}_t \tag{90}$$

Thus, we reproduce the sampler of I2SB.



(a) γ_t.

(b) Ablation study on the shape of γ_t.

Figure 7: Ablation study on the shape of γ_t.

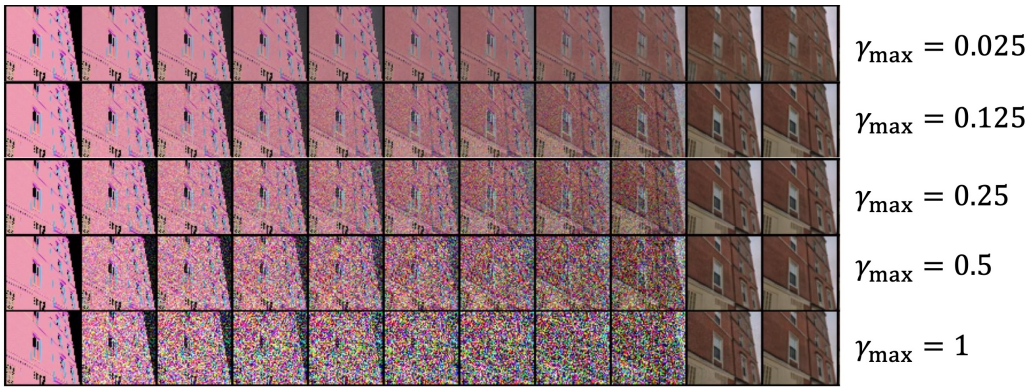


Figure 8: Sampling paths with dfferent choices of γ_t. As γ_t extreamly low, e.g, $\gamma_{\max} = 0.025$, the model will be failed to construct details of images.

E Additional design guideline

α_t and β_t. Theoretically, α_t and β_t can be freely designed, and future work may explore alternative design choices. However, in this paper, we focus on the simple case where $\alpha_t = 1-t$ and $\beta_t = t$. The rationale is as follows: consider the scenario where $\alpha_t = 1 - \beta_t$, which represents an interpolation along the line segment between x_0 and x_1. For the path $p_t^{(1)}(x) = \mathcal{N}((1-\beta_t)x_0 + \beta_t x_1, \gamma_t^2 \mathbb{I})$, where β_t is invertible, it is straightforward to construct another path $p_t^{(2)}(x) = \mathcal{N}((1-t)x_0 + t x_1, \gamma^2_{\beta_t^{-1}} \mathbb{I})$, which achieves the same objective function but uses a different distribution of t during training. Based on this equivalence, setting $\alpha_t = 1 - t$ and $\beta_t = t$ is a reasonable choice.

The shape of γ_t. We conducted an ablation study on γ_t with different shapes. Specifically, we assumed γ_t has the form $\gamma_t = 2\gamma_{\max}\sqrt{t^k(1-t^k)}$, as shown in Fig. 7, γ_t will have different shape as we set different k. The results indicate that the best performance is achieved when $k = 1$, which is the exact setting used in this paper.

$\gamma_{\max}$. Our ablation studies on $\gamma_{\max}$ demonstrate that the optimal values of $\gamma_{\max}$ are approximately 0.125 or 0.25. Furthermore, the sampling paths corresponding to different choices of γ_t are shown in Fig. 8. Adding an appropriate amount of noise to the transition kernel helps in constructing finer details.

ϵ_t. We use the setting $\epsilon_t = \eta\left(\gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2\right)$. The ablation studies on ϵ_t demonstrate that the optimal choice of η for the DDBM-VP model is approximately 0.3, while the best choice for the ECSI model with a Linear Path is around 1.0. Additionally, we present sample paths and generated images under different η settings to illustrate heuristic parameter tuning techniques. The results are shown in Figures 10, 11, and 12. Too small a value of η results in the loss of high-frequency information, while too large a value of η produces over-sharpened and potentially noisy sampled images.

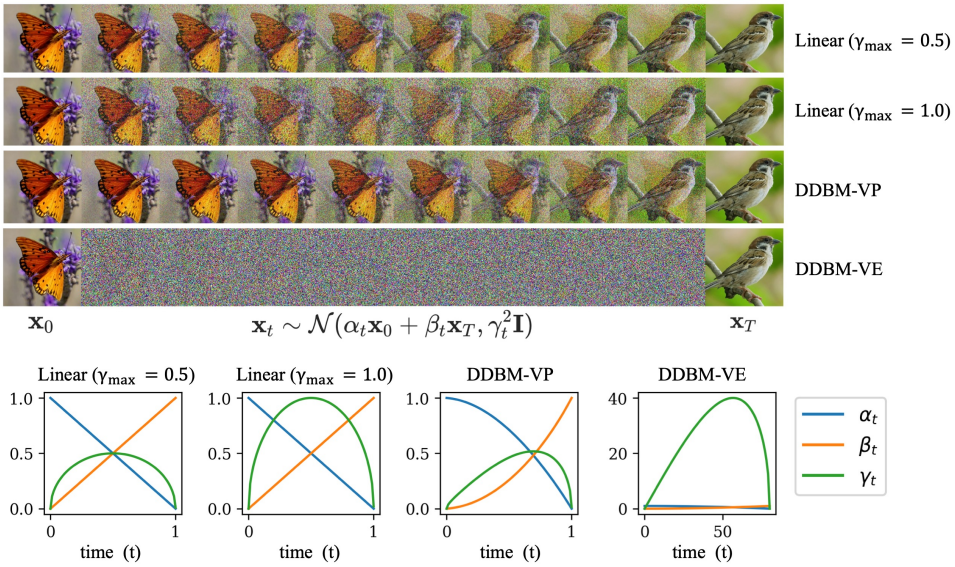


Figure 9: An illustration of design choices of transition kernels and how they affect the I2I translation process. α_t and β_t define the interpolation between two images, while γ_t controls the noise added to the process. ntuitively, the DDBM-VE model introduces excessive noise in the middle stages, which is unnecessary for effective image translation and may explain its poor performance. In contrast, our Linear path results in a symmetrical noise schedule, ensuring a more balanced process. On the other hand, the DDBM-VP path adds more noise near $\mathbf{x}_T$, , indicating that during training, more computational resources are focused around $\mathbf{x}_0$.

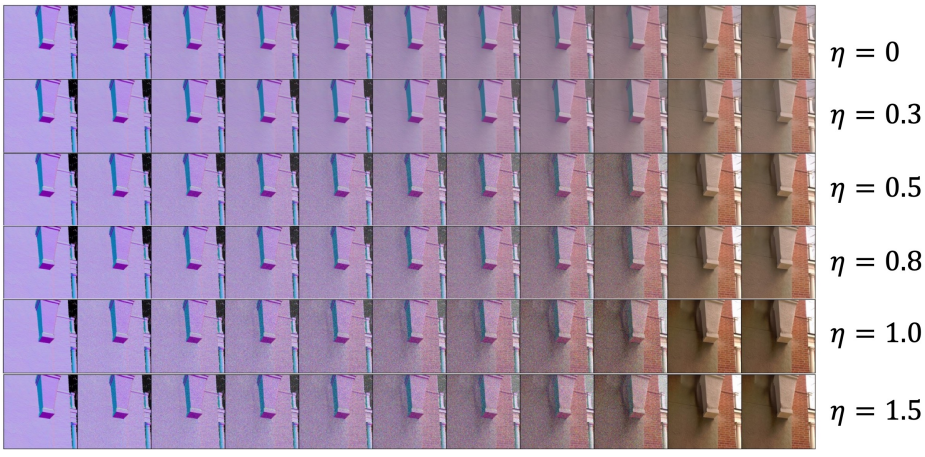


Figure 10: Sampling path with dfferent choices of ϵ_t. As $\epsilon_t = 0$, the generated images lack details, as ϵ_t too large, the sampled images are over-sharpening. The best choices of ϵ_t are around $\epsilon_t = 0.8$ and $\epsilon_t = 1.0$.

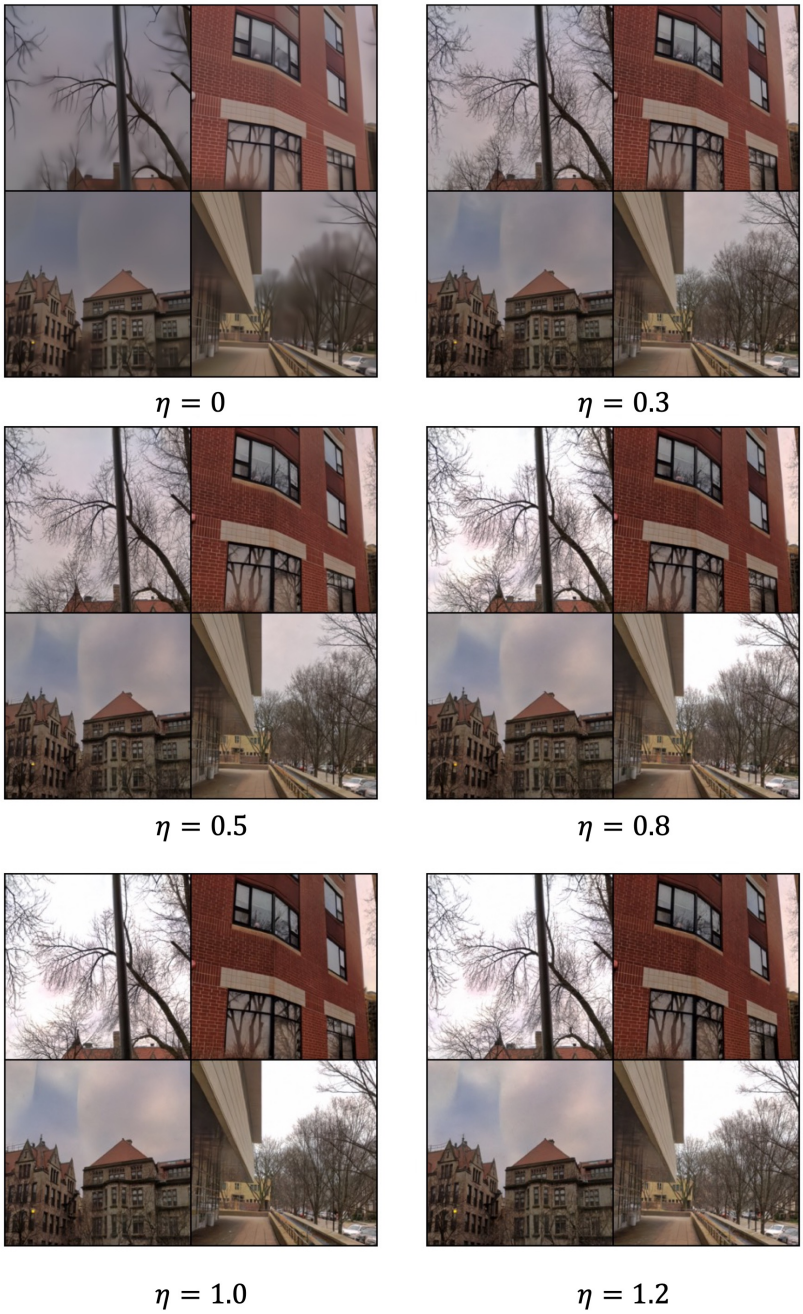


Figure 11: Comparison of sampled images with different ϵ_t for ECSI model, where $\epsilon_t = \eta(\gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2)$, $\gamma_{\max} = 0.25$, $b = 0$.

F Impact Statement

Our method can improve image translation and solving inverse problem, which may benefit applications in medical imaging. However, it is important to note that as with many generative and restoration models, our method could be misused for malicious image manipulation.

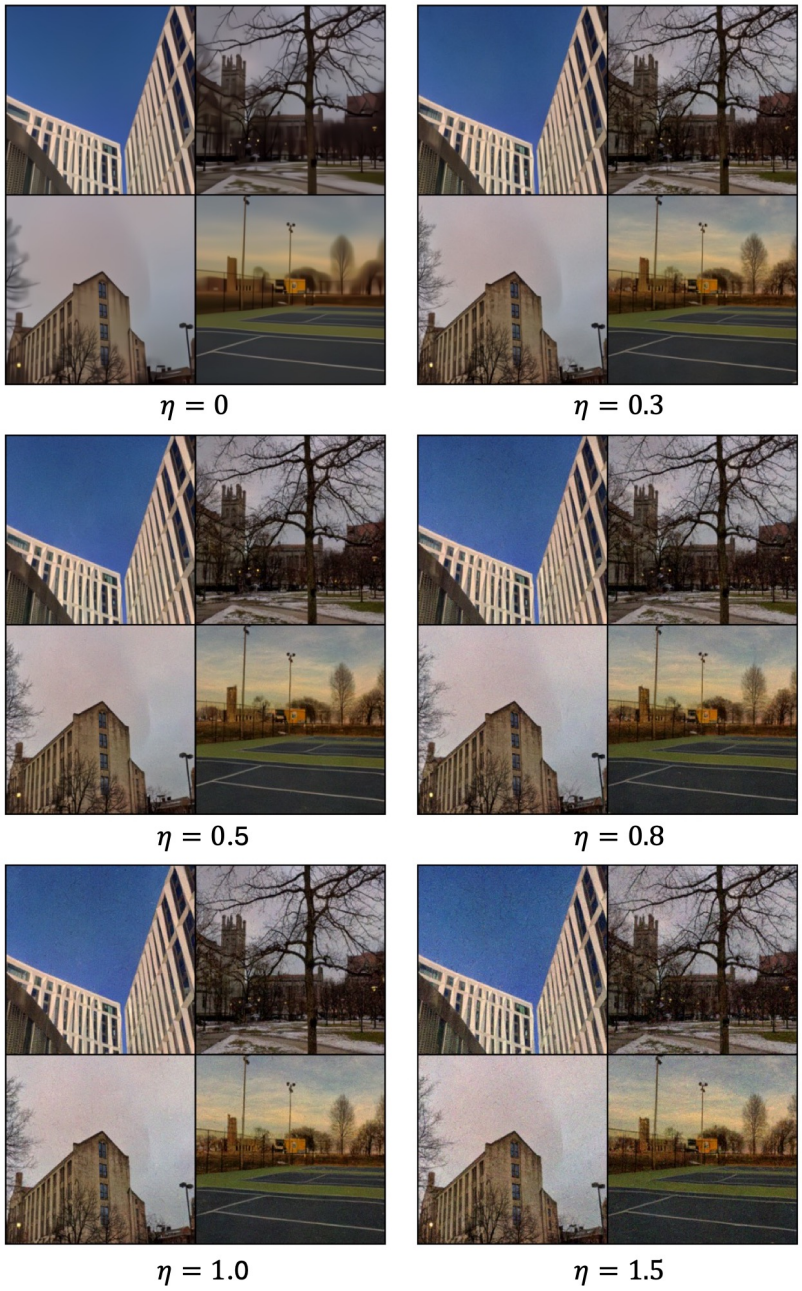


Figure 12: Comparison of sampled images with different ϵ_t for DDBM-VP pretrained model, where $\epsilon_t = \eta(\gamma_t\dot{\gamma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\gamma_t^2)$.

G Experiment Details

Architecture. We maintain the architecture and parameter settings consistent with [18], utilizing the ADM model [10] for 64×64 resolution, modifying the channel dimensions from 192 to 256 and reducing the number of residual blocks from three to two. Apart from these changes, all other settings remain identical to those used for 64×64 resolution.

Training. We include additional pre- and post-processing steps: scaling functions and loss weighting, the same ingredient as [17]. Let $D_\theta(\mathbf{x}_t, \mathbf{x}_T, t) = c_{\text{skip}}(t)\mathbf{x}_t + c_{\text{out}(t)}(t)F_\theta(c_{\text{in}}(t)\mathbf{x}_t, c_{\text{noise}}(t))$,

where F_θ is a neural network with parameter θ, the effective training target with respect to the raw network F_θ is: $\mathbb{E}_{\mathbf{x}_t,\mathbf{x}_0,\mathbf{x}_T,t}\left[\lambda\|c_{\text{skip}}(\mathbf{x}_t + c_{\text{out}}F_\theta(c_{\text{in}}\mathbf{x}_t, c_{\text{noise}}) - \mathbf{x}_0\|^2\right]$. Scaling scheme are chosen by requiring network inputs and training targets to have unit variance $(c_{\text{in}}, c_{\text{out}})$, and amplifying errors in F_θ as little as possible. Following reasoning in [18],

$$c_{\text{in}}(t) = \frac{1}{\sqrt{\alpha_t^2\sigma_0^2 + \beta_t^2\sigma_T^2 + 2\alpha_t\beta_t\sigma_{0T} + \gamma_t^2}}, \quad c_{\text{skip}}(t) = (\alpha_t\sigma_0^2 + \beta_t\sigma_{0T}) * c_{\text{in}}^2, \tag{91}$$

$$c_{\text{out}}(t) = \sqrt{\beta_t^2\sigma_0^2\sigma_1^2 - \beta_t^2\sigma_{0T}^2 + \gamma_t^2\sigma_0^2}c_{\text{in}}, \quad \lambda = \frac{1}{c_{\text{out}}^2}, \quad c_{\text{noise}}(t) = \frac{1}{4}\log(t), \tag{92}$$

where σ_0^2, σ_T^2, and σ_{0T} denote the variance of $\mathbf{x}_0$, variance of $\mathbf{x}_T$ and the covariance of the two, respectively.

We note that TrigFlow [23], adopts the same score reparameterization and pre-conditioning techniques. It can be considered a special case of our framework by setting $\alpha_t = \cos(t)$, $\beta_t = 0$, $\gamma_t = \sigma_0\sin(t)$, $t \in [0, \frac{\pi}{2}]$. In this case, $\sigma_T = 0$, $\sigma_{0T} = 0$,

$$c_{\text{in}}(t) = \frac{1}{\sqrt{\alpha_t^2\sigma_0^2 + \gamma_t^2}} = \frac{1}{\sqrt{\sin^2(t)\sigma_0^2 + \cos^2(t)\sigma_0^2}} = \frac{1}{\sigma_0}, \tag{93}$$

$$c_{\text{skip}}(t) = (\alpha_t\sigma_0^2)c_{in}^2 = \cos(t)\cdot\sigma_0^2\cdot\frac{1}{\sigma_0^2} = \cos(t), \tag{94}$$

$$c_{out}(t) = \sqrt{\gamma_t^2\sigma_0^2}\cdot c_{in} = \sin(t)\sigma_0, \tag{95}$$

$$D_\theta(x_t, t) = c_{\text{skip}}x_t + c_{\text{out}}F_\theta(c_{\text{in}}x_t, c_{\text{noise}}) = \cos(t)x_t + \sin(t)\sigma_0 F_\theta(\frac{1}{\sigma_0}, c_{\text{noise}}). \tag{96}$$

Then we recover TrigFlow.

In our implementation, we set $\sigma_0 = \sigma_T = 0.5, \sigma_{0T} = \sigma_0^2/2$ for all training sessions. Other setting are shown in Table 9.

Table 9: Training settings

Model	Dataset	edges→handbags	edges→handbags	edges→handbags
	η	0	0	0.5
	$\gamma_{\max}$	0.125	0.25	0.125
Setting	GPU	1 A6000 48G	1 H100 96G	1 H100 96G
	Batch size	32	128	200
	Learning rate	1×10^{-5}	5×10^{-5}	1×10^{-4}
	epochs	2078	2106	1443
	Training time	42 days	8 days	11 days
Model	Dataset	DIODE (256×256)	DOIDE (256×256)	
	η	0	0	
	$\gamma_{\max}$	0.125	0.25	
Setting	GPU	1 H100 96G	1 H100 96G	
	Batch size	16	16	
	Learning rate	2×10^{-5}	2×10^{-5}	
	epochs	2617	1745	
	Training time	17 days	25 days	

Sampling. We use the same timesteps distributed according to EDM [17]: $(t_{\max}^{1/\rho} + \frac{i}{N}(t_{\min}^{1/\rho} - t_{\max}^{1/\rho}))^\rho$, where $t_{\min} = 0.001$ and $t_{\max} = 1 - 10^{-4}$. The best performance achieved by setting $\rho = 0.6$ for Edges2handbags and $\rho = 0.8$ for DIODE datasets.

H Licenses

- Edges→Handbags [16]: BSD license.
- DIODE-Outdoor [37]: MIT license.

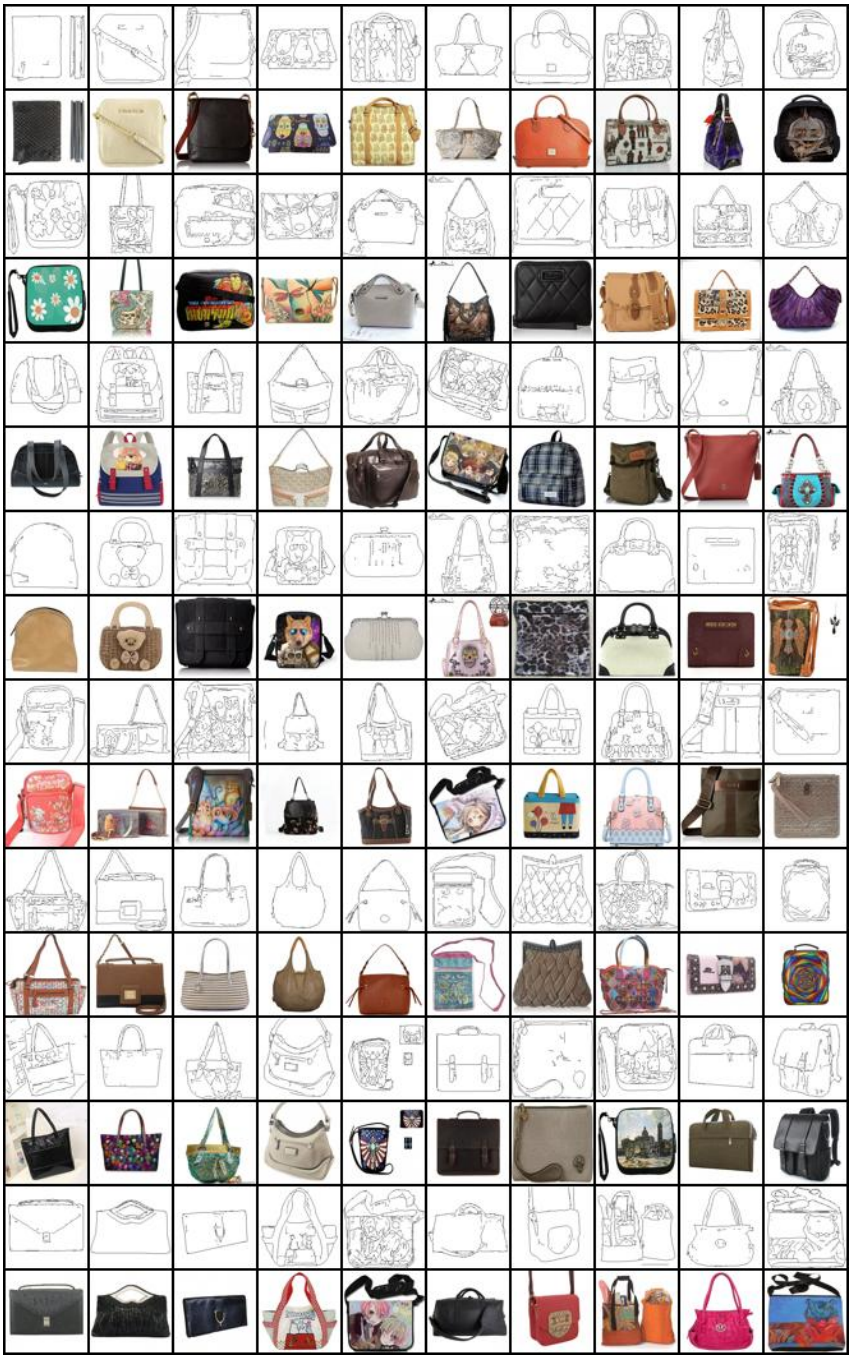


Figure 13: ECSI model and sampler ($\gamma_{\max} = 0.125$, $\eta = 1$, $b = 0$, NFE=5, FID=0.89).

I Additional visualizations

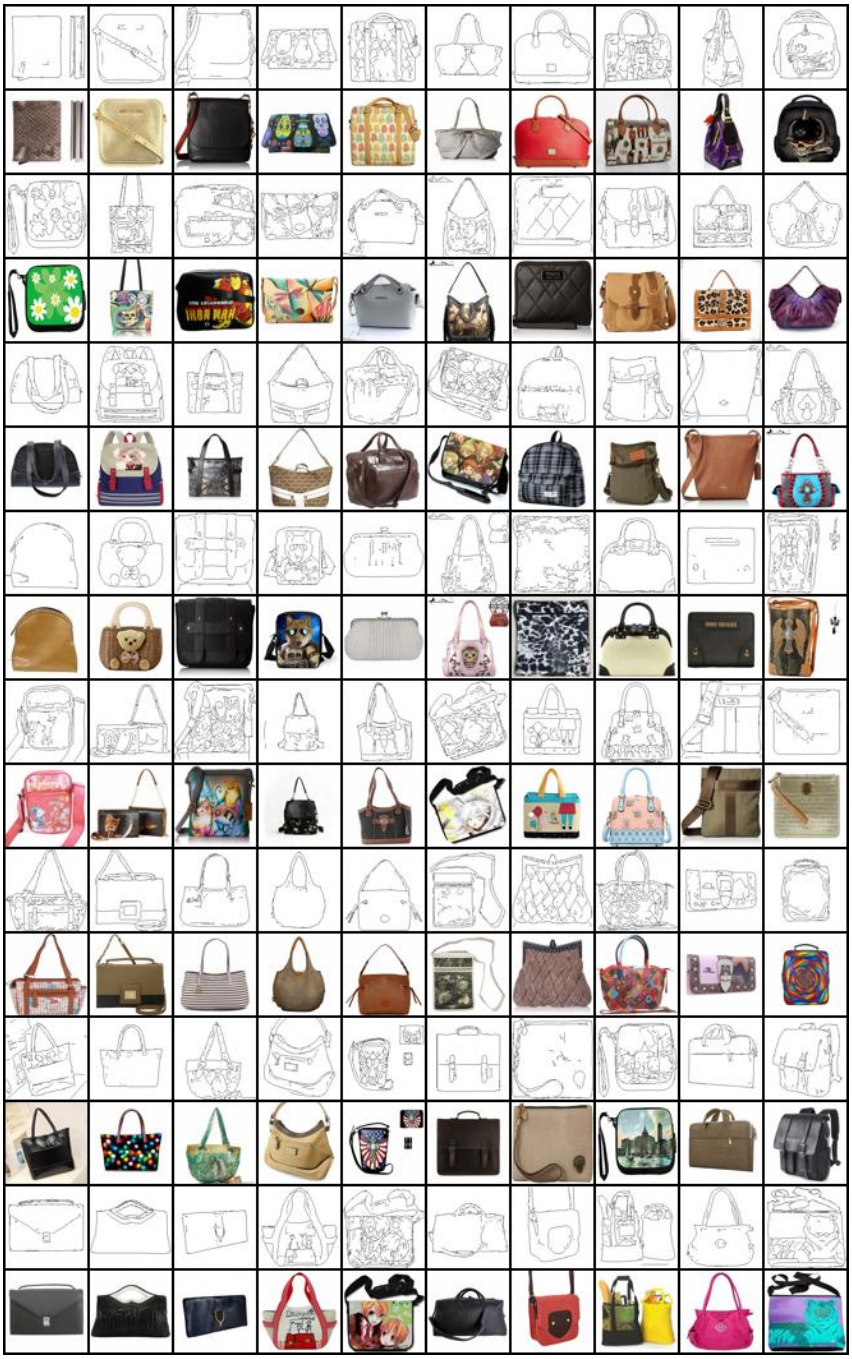


Figure 14: DDBM model and Our sampler (NFE=20, FID=1.53).

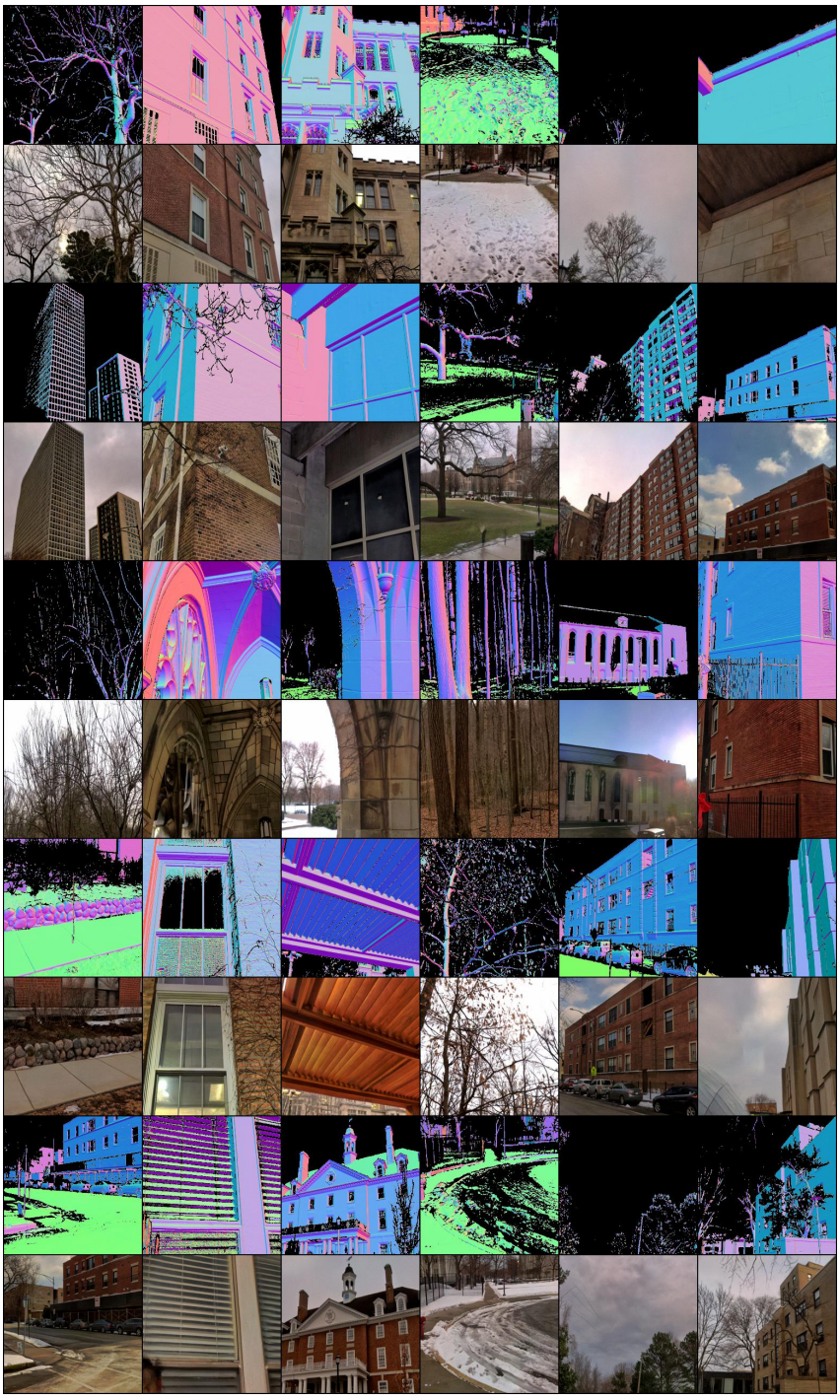


Figure 15: DDBM model and ECSI sampler ($\eta = 0.3$, NFE=20, FID=4.12). Samples for DIODE dataset (conditoned on depth images).

Figure 16: ECSI model and sampler ($\gamma_{\max} = 0.25$, $\eta = 1.0$, $b = 0$, NFE=5, FID = 4.16).

Figure 17: ECSI model and sampler ($\gamma_{\max} = 0.25$, $\eta = 1.0$, $b = 0$, NFE=20, FID = 3.27).

Figure 18: DDBM model and DBIM sampler (NFE=10, FID = 2.46, AFD=5.20).

Figure 19: DDBM model and sampler (NFE=118, FID = 1.83, AFD=6.99).

Figure 20: ECSI model and sampler ($\gamma_{\max} = 0.125$, $b = 1.0$, NFE=10, FID = 2.07, AFD=9.35).

Figure 21: DDBM model and ECSI sampler on 446 test images. (NFE=20, FID = 52.01, AFD=5.60).

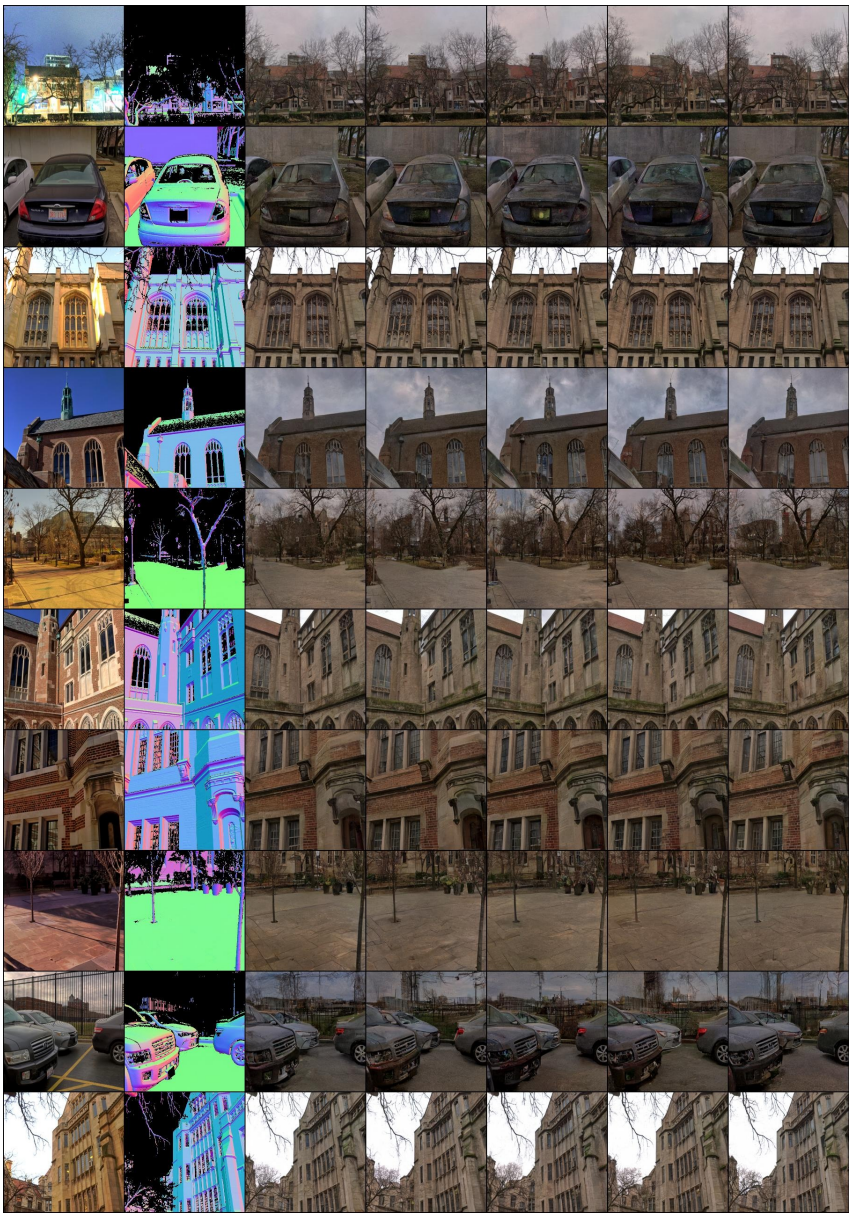


Figure 22: ECSI model and sampler on 446 test images. ($\gamma_{\max} = 0.125$, $b = 0.5$, NFE=20, FID = 55.93, AFD=7.39).

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We ensure that the abstract and introduction clearly summarize the proposed method.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We discuss the limitations in the Conclusion in Sec. 8. For example, we acknowledge that the optimal paths may vary from one scenario to another, indicating a rich avenue for further exploration and refinement in future work.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We include proofs in App. C.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We include detailed experiment information in Sec. 6 and App. G.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We include anonymous code access in the Abstract.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We include experiments in Sec. 6 and additional experiment details in App. G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Not include in the current version. We acknowledge that formal error bars or statistical tests (e.g., t-tests) are not included in the current draft.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include sufficient information on the computer resources in App. G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully reviewed the NeurIPS Code of Ethics and confirm that our research adheres to all its principles.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both the potential positive and negative societal impacts in App. F.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We use standard, publicly available datasets (e.g., Edges2handbags, DIODE, Imagenet), and the risk of misuse is minimal. As such, no specific safeguards were deemed necessary.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We include Licenses section in App. H.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce new code for our proposed method. We include access of an anonymous repository in the Abstract.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our work does not involve any human subjects, user studies, or crowdsourcing experiments. All experiments are conducted using synthetic or publicly available datasets without human participation.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our research does not involve human subjects, user studies, or crowdsourced participation. Therefore, no risks were incurred, and IRB approval was not required.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Large language models (LLMs) were not used in any way that affects the core methodology, experiments, or results presented in this paper. Any assistance from tools such as ChatGPT was limited to minor writing edits or formatting suggestions, which do not influence the scientific contributions or conclusions of the work.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.