# JUMPSAMPLER: MAKING DISTRIBUTION TAILS EASIER TO CHASE

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

## ABSTRACT

Real-world data distributions can exhibit heavy-tailed characteristics in both pixel and latent spaces, assigning non-negligible probability mass to rare or low-density regions. However, diffusion and flow-based generative models rely predominantly on local, continuous sampling dynamics, which can be insufficient for capturing such heavy-tailed structures, potentially leading to under-represented tail events, reduced sample diversity, and distorted distributional statistics. In this paper, we provide a theoretical characterization of this limitation. Specifically, we show that, under standard regularity conditions, most samplers produce light-tailed terminal distributions, whereas sparse heavy-tailed perturbations can induce polynomial tail behavior while keeping the deviation from the original sampler controlled. Therefore, we design **JumpSampler**, a training-free and plug-in sampling strategy that augments pretrained diffusion and flow models with jumps, enabling occasional nonlocal transitions while remaining broadly compatible with diverse sampling schemes and generative architectures. The key design is to inject heavy-tailed perturbations only at sparse sampling steps and progressively anneal their effect over the sampling trajectory, allowing the sampler to explore distant low-density regions while preserving local refinement near the end of generation. We theoretically show that such sparse heavy-tailed jumps induce polynomial tail behavior in the resulting distribution, and further establish stability guarantees showing that the jump-augmented sampler remains controlled in both total variation and Wasserstein distance from the original sampling process. Extensive experiments demonstrate that JumpSampler improves tail fitting, distributional matching, and overall generation quality across both stochastic and deterministic samplers.

## 1 INTRODUCTION

Diffusion and flow-based generative models (Song & Ermon, 2019; Lipman et al., 2022) have emerged as powerful paradigms for high-quality data generation, achieving remarkable performance in image synthesis (Dhariwal & Nichol, 2021), text-to-image generation (Rombach et al., 2022), and other high-dimensional generative tasks (Woo et al., 2024; de Oliveira et al., 2025). At inference time, these models are instantiated through numerical samplers that discretize the learned reverse dynamics into a sequence of finite updates. Stochastic samplers, such as Euler–Maruyama and predictor–corrector schemes (Song et al., 2020b), combine local drift updates with Gaussian perturbations, whereas deterministic samplers, including DDIM (Song et al., 2020a), Euler or Heun schemes (Karras et al., 2022), and specialized high-order solvers such as DPM-Solver (Lu et al., 2022), progressively evolve samples through evaluations of the learned score or velocity field. Despite differences in stochasticity, discretization order, and numerical efficiency, these samplers fundamentally construct the sampling trajectory through successive local updates. This local and incremental samplers is highly effective, but it provides no explicit mechanism for occasional nonlocal transitions across the state space.

Nonlocal behavior may be useful when modeling distributions with substantial probability mass in low-density or extreme regions. Heavy-tailed and scale-free statistics arise broadly in empirical data (Clauset et al., 2009), and natural images exhibit several pronounced departures from Gaussian behavior. Early studies showed that local image statistics, such as contrast, are strongly non-Gaussian and approximately scale invariant (Ruderman & Bialek, 1993), while spatial-frequency statistics of natural scenes exhibit approximate power-law scaling (Ruderman, 1997). They nevertheless suggest that Gaussian statistics alone may provide an incomplete description of the low-density structures



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

present in natural data. Related evidence has also emerged in latent space. Recent works have incorporated heavy-tailed distributions into pixel and latent spaces, including $\alpha$-stable Lévy processes for score-based generation (Yoon et al., 2023) and Student's $t$ distributions for VAE latent modeling (Kim et al., 2024; Bouayed et al., 2025). Other studies explicitly regularize latent representations toward Gaussian statistics (Hwang et al., 2025), further suggesting that Gaussianity should not be taken for granted in learned representations. These results indicate that Gaussian-like tails can be restrictive for modeling low-density or extreme regions. However, accurately capturing such regions is particularly important when they correspond to extreme events, whose statistics can be substantially distorted when generative models fail to extrapolate the distribution tails (Stamatelopoulos & Sapsis, 2025).

This motivates us to revisit the stochastic dynamics underlying conventional diffusion sampling. Most diffusion processes are driven by Brownian motion and therefore rely on Gaussian, light-tailed perturbations; as proved in Theorem 1, under standard regularity conditions, such dynamics tend to preserve light-tailed behavior in the generated distribution. Recent studies have also pointed out the resulting difficulty in approximating heavy-tailed target distributions (Fassina et al., 2026). In contrast, Lévy and jump processes have long been used in stochastic modeling to capture non-Gaussian, heavy-tailed phenomena and abrupt large-scale variations that are difficult to represent with Brownian dynamics (Merton, 1976; Barndorff-Nielsen & Shephard, 2002). Their ability to naturally accommodate heavy tails and occasional large transitions therefore provides a promising mechanism for improving the approximation of heavy-tailed target distributions.

To incorporate these desirable properties into existing samplers, we design **JumpSampler**, which builds on a base sampler by injecting stochastic jumps into its sampling trajectory. Theoretically, Theorem 2 establishes that JumpSampler induces heavy-tailed sample distributions through heavy-tailed jump perturbations. Furthermore, Theorems 3 and 4 provide distributional and mean-square stability guarantees, respectively, showing that JumpSampler remains close to the original sampling dynamics under controlled jump rates. Empirically, we show that JumpSampler is broadly compatible with both stochastic and deterministic samplers and consistently improves generation quality across multiple generation tasks. Our contributions are summarized as follows:

- **Heavy-Tail Perspective on Generative Sampling:** We analyze the limitations of conventional diffusion and flow samplers in capturing such behavior. In particular, under standard regularity conditions, commonly used continuous sampling dynamics tend to produce light-tailed terminal distributions, revealing a structural mismatch with heavy-tailed targets.
- **JumpSampler with Theoretical Guarantees:** We design JumpSampler, a training-free and plug-in framework that augments pretrained samplers with sparse, annealed, heavy-tailed jumps while preserving the original dynamics at most inference steps. We theoretically show that the proposed jump augmentation can induce heavy-tailed behavior in the resulting distribution, while establishing quantitative stability guarantees that bound its deviation from the original sampler.
- **Broad Empirical Validation:** We evaluate JumpSampler across class-conditional and text-to-image generation with diverse diffusion and flow models, covering both stochastic and deterministic sampling schemes. Extensive experiments demonstrate consistent improvements in tail fitting, distributional matching, and overall generation quality, highlighting the effectiveness and broad applicability of JumpSampler.

## 2 Background

### 2.1 General Diffusion Models

Consider a continuous-time forward diffusion process (Song & Ermon, 2019)

$$\mathrm{d}x_t = f(x_t, t)\mathrm{d}t + g(t)\mathrm{d}w_t, \tag{1}$$

where $f(x_t, t)$ denotes the drift coefficient, $g(t)$ denotes the diffusion coefficient, and $w_t$ is a standard Wiener process.

The corresponding reverse-time SDE (Anderson, 1982) is

$$\mathrm{d}x_t = \left[f(x_t, t) - g^2(t)\nabla_x \log p_t(x_t)\right]\mathrm{d}t + g(t)\mathrm{d}\bar{w}_t, \tag{2}$$

where $\bar{w}_t$ is a reverse-time Wiener process and $p_t(x)$ denotes the marginal density at time $t$.

In practice, the score function is approximated by a pretrained neural network (Song et al., 2020b): $s_\theta(x_t, t) \approx \nabla_x \log p_t(x_t)$. The learned reverse-time SDE can therefore be expressed as

$$\mathrm{d}x_t = \left[f(x_t, t) - g^2(t) s_\theta(x_t, t)\right] \mathrm{d}t + g(t) \mathrm{d}\bar{w}_t.$$

Deterministic probability-flow ODEs (Lipman et al., 2022) and rectified-flow models (Liu et al., 2022; Liu, 2022) can similarly be written as

$$\mathrm{d}x_t = v_\theta(x_t, t) \mathrm{d}t,$$

where $v_\theta(x_t, t)$ denotes the learned velocity field.

### 2.2 Jump-Diffusion Processes

Classical diffusion processes evolve continuously and are locally driven by Brownian perturbations. A natural extension is a *jump-diffusion process*, which augments the continuous dynamics with a discontinuous jump component. A general jump-diffusion process (Hanson, 2007) takes the form

$$dX_t = b(X_{t^-}, t)\, dt + \sigma(t)\, dW_t + dJ_t, \tag{3}$$

where $X_{t^-}$ denotes the state immediately before time $t$, and $J_t$ is a jump process. Generally, the jump component can be represented through a Poisson random measure $N(dt, dz)$:

$$dX_t = b(X_{t^-}, t)\, dt + \sigma(t)\, dW_t + \int_{\mathbb{R}^d} z\, N(dt, dz), \tag{4}$$

where the intensity measure determines both the arrival rate and the distribution of jump sizes. For a compound Poisson process with time-dependent rate $r(t)$ and jump law $\mu_t$, the corresponding Lévy measure is $\nu_t(dz) = r(t)\mu_t(dz)$.

If $p_t(x)$ denotes the probability density of $X_t$, its evolution is governed by the nonlocal Fokker–Planck equation (Applebaum, 2009; Cont & Tankov, 2003)

$$\partial_t p_t(x) = -\nabla \cdot (b(x, t) p_t(x)) + \frac{1}{2}\sigma^2(t) \Delta p_t(x) + \int_{\mathbb{R}^d} \left[p_t(x - z) - p_t(x)\right] \nu_t(dz).$$

The first two terms describe the conventional local drift–diffusion dynamics, whereas the integral term transports probability mass nonlocally through jumps. When $\sigma \equiv 0$, Equation 3 reduces to a deterministic flow augmented with stochastic jumps, allowing the same formulation to cover both stochastic diffusion samplers and deterministic flow-based samplers. We defer a more detailed discussion of related work to Appendix A.

## 3 Method

### 3.1 Motivation

Standard diffusion (Sohl-Dickstein et al., 2015) and flow (Liu et al., 2022; Lipman et al., 2022) samplers are primarily driven by Gaussian initialization and deterministic local transformations induced by the learned score or velocity field, with additional Gaussian noise for stochastic samplers. Under regularity conditions on these transformations, a finite composition of such Gaussian-driven updates retains strong concentration, yielding a terminal distribution whose radial tail is lighter than any polynomial. We formalize this observation in Theorem 1.

Latent diffusion models (Rombach et al., 2022), however, typically operate on representations produced by a KL-regularized variational autoencoder (VAE, Pinheiro Cinelli et al. (2021)). Although the VAE objective encourages the conditional posterior $q_\phi(z|x)$ to remain close to a Gaussian prior, this does not necessarily imply that the aggregated posterior $q_\phi(z) = \int q_\phi(z|x) p_{\mathrm{data}}(x)\, \mathrm{d}x$ is itself light-tailed. The following proposition makes this distinction explicit.

**Proposition 1** (Gaussian KL regularization does not guarantee light-tailed aggregated latents)**.** *There exists a family of conditional latent distributions for which the average KL divergence to a standard Gaussian prior can be made arbitrarily small, while the corresponding aggregated latent distributions remain polynomially heavy-tailed.*

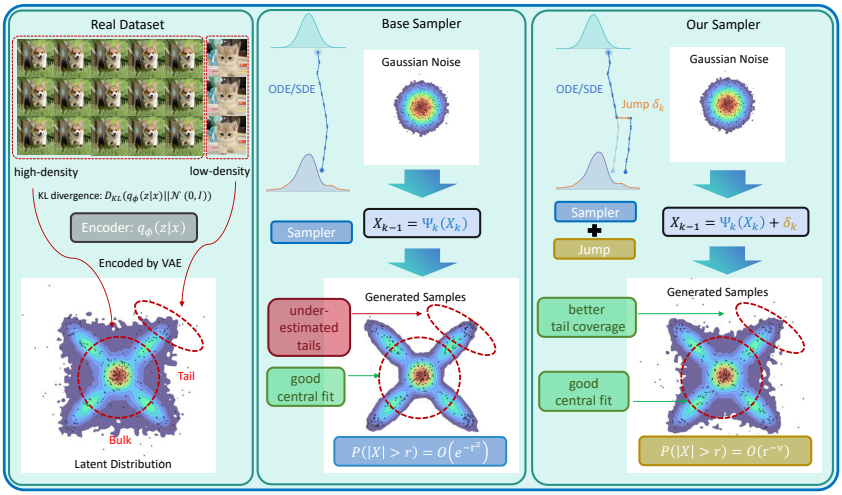



Figure 1: **Overview of JumpSampler.** Real data can exhibit low-density tail regions in latent space that are difficult to reach using purely local sampling dynamics. Starting from the same Gaussian prior, the base sampler evolves only through local ODE/SDE updates and may underestimate such tails, whereas JumpSampler introduces sparse heavy-tailed jumps to enable occasional nonlocal transitions while retaining the original local refinement process.

*Proof.* See Appendix B.1. □

Together, Proposition 1 and Theorem 1 reveal a potential tail mismatch: Gaussian KL regularization does not exclude polynomially heavy-tailed latent targets, whereas the Gaussian-driven base sampler remains lighter than every polynomial under the stated regularity conditions. This motivates augmenting the pretrained sampling dynamics with sparse heavy-tailed perturbations.

### 3.2 Jump-Augmented Sampling Dynamics

Based on the motivation above, we propose **JumpSampler**, a training-free sampling modification that augments pretrained diffusion and flow models with sparse, annealed, and heavy-tailed perturbations. As illustrated in Figure 1, standard samplers evolve from the Gaussian prior through a sequence of local updates and may therefore under-represent low-density tail regions. JumpSampler preserves these original sampling dynamics at most inference steps, while introducing occasional nonlocal transitions through sparse jumps.

**Continuous-time formulation.** Standard diffusion and flow samplers generate samples through a sequence of local updates determined by the learned score or velocity field. Although such local dynamics are effective for progressively refining a noisy state, they may have limited ability to perform occasional nonlocal transitions during sampling. To introduce such transitions, we augment the reverse sampling dynamics with a stochastic jump component:

$$\mathrm{d}X_t^J = b_t(X_t^J)\mathrm{d}t + \sigma_t \mathrm{d}\bar{w}_t + \mathrm{d}\bar{J}_t. \tag{5}$$

The jump component is designed to perturb the original sampling trajectory only at a small number of randomly selected times. To distinguish the occurrence of a jump from its displacement, we represent the jump process as

$$\bar{J}_{t:T} = \sum_{i:\,\tau_i \in [t,T]} a(\tau_i)\rho_i u_i, \tag{6}$$

where $\{\tau_i\}$ denotes the set of activated jump times generated by Poisson process with time-dependent intensity $r(t)$. For a jump occurring at time $\tau_i$, the deterministic factor $a(\tau_i)$ specifies the time-dependent scale of the perturbation, while $\rho_i \sim |\mathrm{StudentT}(\nu)|$ specifies its random radial magnitude.

The perturbation direction is given by

$$u_i = \frac{z_i}{\|z_i\|_2}, \qquad z_i \sim \mathcal{N}(0, I_d),$$

so that $\|u_i\|_2 = 1$. Specifically, we define the intensity and magnitude as

$$r(t) = \frac{\lambda}{T}\left(\frac{t}{T}\right)^{\beta}, \quad a(t) = \left(\frac{t}{T}\right)^{\beta}, \tag{7}$$

where $\lambda > 0$ and $\beta > 0$. Under the convention that $t = T$ corresponds to the initial high-noise state and $t = 0$ corresponds to the final generated sample, the intensity $r(t)$ is larger at the beginning of sampling and gradually decreases toward the terminal stage. Consequently, jumps are more likely to occur while the state remains highly noisy and become increasingly unlikely as the sample is refined.

**Discrete implementation.** We next discretize the jump mechanism on the inference-time grid

$$0 = t_0 < t_1 < \cdots < t_K = T.$$

At the $k$-th step, JumpSampler first performs the update of the original pretrained sampler, then adds a sparse heavy-tailed perturbation to the updated state:

$$X_{k-1}^J = \Psi_k(X_k^J) + \delta_k. \tag{8}$$

The discrete jump perturbation is defined as (Zheng et al., 2024)

$$\delta_k = B_k a(t_k)\rho_k u_k = \begin{cases} a(t_k)\rho_k u_k, & \text{with probability } p_k, \\ 0, & \text{with probability } 1 - p_k. \end{cases} \tag{9}$$

Here, $B_k \sim \text{Bernoulli}(p_k)$ determines whether a jump is activated at the $k$-th inference step. Conditioned on $B_k = 1$, $u_k$ and $\rho_k$ are sampled as defined above. Moreover, $p_k$ can be approximated by evaluating the intensity at $t_k$:

$$p_k^{\text{Poisson}} \approx r(t_k)h_k, \quad h_k = t_k - t_{k-1}. \tag{10}$$

Equation 10 is a first-order discretization of the underlying Poisson process. Under a coarse inference schedule, however, this approximation may produce an excessively large activation probability. We therefore clip it by a prescribed upper bound:

$$p_k = \min\left\{p_k^{\text{Poisson}}, p_{\max}\right\}. \tag{11}$$

The resulting update preserves the original pretrained sampler whenever $B_k = 0$, while introducing a time-annealed heavy-tailed displacement whenever $B_k = 1$. Since no additional evaluation of the score or velocity network is required, the jump operator can be inserted after each base-solver update without modifying the pretrained model. The complete sampling procedure is summarized in Algorithm 1. **Unless otherwise specified, we always set $\nu = 3$, $p_{\max} = 1.0$.**

## 4 Theoretical Analysis

Throughout the theoretical analysis, we study the unclipped discretization $p_k = r(t_k)h_k$, corresponding to the standard sufficiently fine-grid regime. When $\nu > 2$, we additionally write $m_2 := \mathbb{E}[\rho_k^2] = \frac{\nu}{\nu-2}$, which will be used in the Wasserstein analysis. By the isotropy and independence of the jump variables, we have $\mathbb{E}[\delta_k] = 0$, $\mathbb{E}\left[\|\delta_k\|_2^2\right] = p_k a_k^2 m_2$.

### 4.1 Euler-Type Base Solver and Regularity

To analyze the effect of jump augmentation independently of a particular sampler, we first introduce a generic Euler-type representation of the pretrained sampling dynamics. We write the $k$-th update of the base sampler as

$$X_{k-1}^0 = \Psi_k(X_k^0; \xi_k) = X_k^0 + h_k F_k(X_k^0) + g_k \xi_k + d_k. \tag{12}$$

**Algorithm 1** JumpSampler

**Require:** Pretrained sampler $\{\Psi_k\}_{k=1}^K$, schedule $\{t_k\}_{k=0}^K$, $\lambda, \beta, p_{\max}, \nu$, conditioning $c$

Sample $x_K$ from the model prior
**for** $k = K, K-1, \ldots, 1$ **do**
  $\widetilde{x}_{k-1} \leftarrow \Psi_k(x_k; c)$
  $p_k \leftarrow \min\left\{\lambda t_k^\beta (t_k - t_{k-1})/T^{\beta+1}, p_{\max}\right\}$, $B_k \sim \text{Bernoulli}(p_k)$
  **if** $B_k = 1$ **then**
    $z_k \sim \mathcal{N}(0, I_d)$, $\rho_k \sim |\,\text{StudentT}(\nu)|$, $u_k \leftarrow z_k/\|z_k\|_2$ ▷ Sample jump magnitude and direction
    $x_{k-1} \leftarrow \widetilde{x}_{k-1} + (t_k/T)^\beta \rho_k u_k$ ▷ Jump Sampler
  **else**
    $x_{k-1} \leftarrow \widetilde{x}_{k-1}$ ▷ Baseline Sampler
  **end if**
**end for**
**return** $x_0$

Here, $F_k : \mathbb{R}^d \to \mathbb{R}^d$ denotes the learned or solver-dependent vector field, $g_k \geq 0$ is a deterministic noise coefficient, $d_k \in \mathbb{R}^d$ is a deterministic offset, and $\xi_k \sim \mathcal{N}(0, I_d)$ represents the Gaussian randomness at the $k$-th step. Deterministic ODE and flow samplers are included as the special case $g_k = 0$. The initial state is

$$X_K^0 = m_K + g_{\text{init}}\xi_{\text{init}}, \qquad \xi_{\text{init}} \sim \mathcal{N}(0, I_d), \tag{13}$$

where all Gaussian variables are mutually independent.

Our analysis requires controlling how a perturbation to the current state propagates through the remaining solver steps. Specifically, we need an upper bound to prevent perturbations from being amplified without control, and, for the heavy-tail analysis, a corresponding lower bound to ensure that a sufficiently large jump is not completely collapsed by subsequent updates. These properties follow from the following regularity conditions.

**Assumption 1** (Global Lipschitz regularity)**.** *For every $k$, the vector field $F_k$ is $M_k$-Lipschitz, and $L_k^+ := 1 + h_k M_k < \infty$.*

**Assumption 2** (Non-collapse condition)**.** *For every $k$, $L_k^- := 1 - h_k M_k > 0$.*

Together, Assumptions 1 and 2 imply the single-step bound

$$L_k^- \|x - y\|_2 \leq \|\Psi_k(x; \xi_k) - \Psi_k(y; \xi_k)\|_2 \leq L_k^+ \|x - y\|_2. \tag{14}$$

The upper bound is to control the overall effect of jump perturbations, while the lower bound ensures that a nondegenerate jump retains a nonzero fraction of its magnitude after subsequent solver steps.

### 4.2 Tail Separation between Base and Jump-Augmented Samplers

We now compare the terminal tail behavior of the base sampler and JumpSampler under the same pretrained dynamics. We first establish the tail behavior induced solely by the Gaussian initialization and Gaussian solver randomness.

**Theorem 1** (Light-tailed terminal distribution of the base sampler)**.** *Under Assumption 1, the terminal state of the base sampler has a sub-Gaussian radial tail. In particular, for every finite $\nu > 0$,*

$$\mathbb{P}\left(\left\|X_0^0\right\|_2 > r\right) = o\left(r^{-\nu}\right). \tag{15}$$

*Proof.* See Appendix B.2. □

We next consider JumpSampler under the same base solver and the same Gaussian randomness. Starting from the same initial state, $X_K^J = X_K^0$, its $k$-th update is

$$X_{k-1}^J = \Psi_k(X_k^J; \xi_k) + B_k a(t_k)\rho_k u_k, \qquad \rho_k \sim |\text{StudentT}(\nu)|\,. \tag{16}$$

**Theorem 2** (Polynomial terminal tail of JumpSampler)**.** *Assume that $0 < p_k < 1$ for every $k$. Under Assumptions 1 and 2, for Equation 16, there exist constants $0 < c_J \leq C_J < \infty$ and $r_J > 0$ such that, for every $r \geq r_J$,*

$$c_J r^{-\nu} \leq \mathbb{P}\left(\left\|X_0^J\right\|_2 > r\right) \leq C_J r^{-\nu}. \tag{17}$$

*Proof.* See Appendix B.5. □

Theorems 1 and 2 therefore establish a clear tail separation between two samplers:

$$\mathbb{P}\big(\|X_0^0\|_2 > r\big) = o(r^{-\nu}), \qquad \mathbb{P}\big(\|X_0^J\|_2 > r\big) = O(r^{-\nu}). \tag{18}$$

Intuitively, trajectories on which no jump occurs retain the light-tailed behavior of the base sampler, whereas trajectories containing a nondegenerate Student's $t$ jump contribute a polynomially decaying tail component. The jump probabilities and annealing coefficients determine how often and when such nonlocal events occur, while the radial jump distribution determines the asymptotic tail exponent.

### 4.3 Distributional Stability

We next quantify the distributional deviation introduced by the jump component, considering total variation and Wasserstein distance stability.

**Total variation stability.** Consider the continuous-time idealization of JumpSampler under an increasing sampling clock $s = T - t$, where $s = 0$ corresponds to the initial high-noise state and $s = T$ corresponds to the final generated sample. Then the diffusion SDEs become

$$\mathrm{d}X_s^0 = b_s(X_s^0)\,\mathrm{d}s + \sigma_s\,\mathrm{d}W_s, \quad \mathrm{d}X_s^J = b_s(X_s^J)\,\mathrm{d}s + \sigma_s\,\mathrm{d}W_s + \mathrm{d}J_s. \tag{19}$$

**Theorem 3** (Total variation stability)**.** *Let $Q_s^0 := \mathcal{L}(X_s^0)$, $Q_s^J := \mathcal{L}(X_s^J)$. For the jump intensity schedule of JumpSampler,*

$$\mathrm{TV}\left(Q_T^J, Q_T^0\right) \le 1 - \exp\left(-\frac{\lambda}{\beta+1}\right) \le \frac{\lambda}{\beta+1}. \tag{20}$$

*Proof.* See Appendix B.6 and Appendix B.7. □

**Wasserstein stability.** Consider the finite-step sampler under a synchronous coupling:

$$X_{k-1}^0 = \Psi_k(X_k^0; \xi_k), \quad X_{k-1}^J = \Psi_k(X_k^J; \xi_k) + \delta_k, \quad X_K^J = X_K^0. \tag{21}$$

**Theorem 4** (Wasserstein stability under synchronous coupling)**.** *Let $Q^0 := \mathcal{L}(X_0^0)$, $Q^J := \mathcal{L}(X_0^J)$. Suppose that $\nu > 2$. Under Assumption 1,*

$$W_2^2(Q^J, Q^0) \le \frac{\lambda m_2}{T} \sum_{k=1}^{K} \left(A_k^+\right)^2 \left(\frac{t_k}{T}\right)^{3\beta} h_k, \quad A_k^+ = \prod_{j=1}^{k-1} L_j^+, A_1^+ = 1. \tag{22}$$

*Proof.* See Appendix B.8. □

**Corollary 1** (Uniform stability under grid refinement)**.** *If there exists a constant $C \ge 0$ such that $L_k^+ \le 1 + Ch_k$ for every $k$, then*

$$\limsup_{\max_k h_k \to 0} W_2^2(Q^J, Q^0) \le \frac{\lambda m_2 \exp(2CT)}{3\beta + 1}. \tag{23}$$

*Proof.* See Appendix B.9. □

The two bounds also clarify the roles of the jump hyperparameters. $\lambda$ and $\beta$ provide direct control over the distributional deviation introduced by JumpSampler. Choosing a smaller $\lambda$ or a larger $\beta$ yields a tighter stability bound and keeps the jump-augmented distribution closer to the base sampler. Importantly, these parameters control the frequency and effective mass of the heavy-tailed component rather than its asymptotic tail exponent: as established in Theorem 2, the terminal tail remains of order $r^{-\nu}$ as long as the jump mechanism is nondegenerate. Thus, appropriate hyperparameter choices allow JumpSampler to introduce heavy-tailed nonlocal behavior while keeping the overall distributional shift quantitatively controlled.

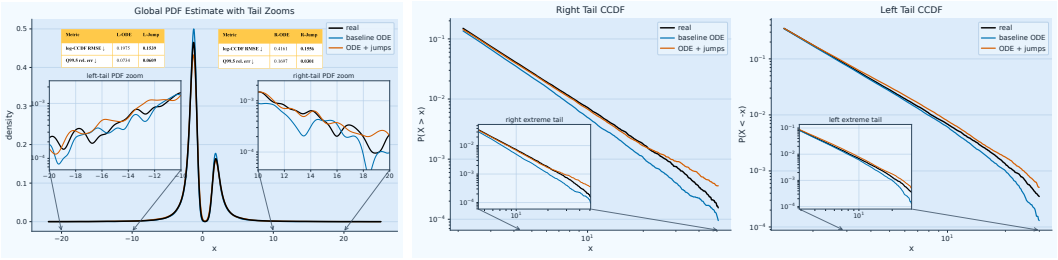



Figure 2: **Heavy-tailed jumps improve tail fitting on a toy target distribution. Left:** Global density estimates with enlarged tail views. **Middle and right:** Log–log CCDFs of the right and left tails. The baseline ODE underestimates tail probabilities, whereas JumpSampler reduces the tail mismatch while preserving a similar fit in the central region.

# 5 Experiments

## 5.1 Toy Example

We first validate JumpSampler on a 1D asymmetric heavy-tailed target constructed from a mixture of truncated Pareto components. We train a flow model with linear interpolation and compare the standard Euler ODE sampler with its jump-augmented counterpart using the same pretrained velocity field. As shown in Figure 2, the baseline accurately fits the high-density region but systematically underestimates both tails, while sparse Student-$t$ jumps substantially improve tail fitting, especially in the extreme-tail regime, with little effect on the distribution bulk. Detailed settings and additional results are provided in Appendix C.1.

## 5.2 Main Results

Table 1: **Quantitative comparison on Class-Conditional tasks.** Comparison of different evaluation metrics on Class-Conditional ImageNet datasets with different diffusion architectures.

| **ImageNet (seed 0)** | | | |
|---|---|---|---|
| **Model-256, 50k samples, 250 inference timesteps** | FID↓ | IS↑ | sFID↓ |
| SiT-XL/2 (REPA)($\omega = 1.35$, SDE) | 1.80 | 284.0 | 4.5 |
| **SiT-XL/2 (REPA) + Ours** ($\beta = 3.0, \lambda = 1.0$) | **1.76** | **284.1** | 4.5 |
| SiT-XL/2 (REPA)($\omega = 1.0$, SDE) | 5.89 | 158.2 | 5.7 |
| **SiT-XL/2 (REPA) + Ours** ($\beta = 4.0, \lambda = 1.0$) | **5.82** | **159.3** | 5.7 |
| SiT-XL/2 (REPA, Interval) ($\omega = 1.8, t_h = 0.7$, SDE) | 1.43 | 305.1 | 4.5 |
| **SiT-XL/2 (REPA, Interval) + Ours** ($\beta = 2.4, \lambda = 1.0$) | **1.39** | **305.7** | 4.6 |
| SiT-XL/2 (REPA, Interval) ($\omega = 1.8, t_h = 0.7$, ODE) | 1.54 | 283.7 | 4.6 |
| **SiT-XL/2 (REPA, Interval) + Ours** ($\beta = 2.5, \lambda = 1.0$) | **1.53** | **284.9** | 4.6 |
| DiT-XL/2 ($\omega = 1.5$, SDE sampler) | 2.29 | 276.8 | 4.6 |
| **DiT-XL/2 + Ours**($\beta = 1.5, \lambda = 1.0$) | **2.24** | **277.8** | **4.6** |

Table 2: **ECS on ImageNet-256, SiT-XL/2.** Raw ECS retains coordinate-wise mean differences, while centered ECS subtracts the mean of each distribution before evaluation.

| | Raw ECS ($\times 10^{-3}$) ↓ | | Centered ECS ($\times 10^{-3}$) ↓ | |
|---|---|---|---|---|
| $T$ | Baseline | Ours | Baseline | Ours |
| 0.02 | 5.282 | **5.278** | 0.0448 | **0.0435** |
| 0.05 | 5.283 | **5.279** | 0.1120 | **0.1088** |
| 0.10 | 5.286 | **5.282** | 0.2241 | **0.2177** |
| 0.20 | 5.292 | **5.286** | 0.4477 | **0.4352** |
| 0.50 | 5.281 | **5.273** | 1.0940 | **1.0677** |
| 0.70 | 5.239 | **5.230** | 1.4845 | **1.4531** |
| 1.00 | 5.129 | **5.118** | 1.9912 | **1.9551** |
| 2.00 | 4.511 | **4.496** | 3.0449 | **2.9955** |
| 3.00 | 3.783 | **3.776** | 3.4083 | **3.3604** |

**Class-conditional image generation.** We evaluate JumpSampler on ImageNet-256 (Deng et al., 2009) with different pretrained architectures and sampling configurations. As shown in Table 1, JumpSampler consistently improves FID (Heusel et al., 2017) across SiT (Ma et al., 2024; Yu et al., 2025) and DiT (Peebles & Xie, 2022)), including SDE sampler (Ho et al., 2020), ODE sampler, (Song et al., 2020a), Classifier-Free Guidance (Ho, 2022) and Interval Guidance (Kynkäänniemi et al., 2024). Meanwhile, IS (Salimans et al., 2016) is generally maintained or improved. These results suggest that the proposed jump mechanism can be incorporated into different pretrained sampling dynamics without retraining while preserving the overall generation quality. Pixel-space results are reported in Appendix C.5.

**Embedded Characteristic Score.** To further evaluate distributional discrepancies beyond Gaussian moment matching, we report the Embedded Characteristic Score (ECS) (Tam & Engelhardt, 2025) in the Inception-v3 feature space (Szegedy et al., 2016), same as FID. **ECS compares the characteristic**

**functions of real and generated feature distributions and is sensitive to discrepancies beyond the first two moments.** Given $d$-dimensional Inception features, we compute

$$\mathrm{ECS}(T) = \frac{1}{d}\sum_{j=1}^{d}\left|\frac{1}{N_r}\sum_{n=1}^{N_r} e^{iTz_{n,j}^{(r)}} - \frac{1}{N_g}\sum_{n=1}^{N_g} e^{iTz_{n,j}^{(g)}}\right|, \tag{24}$$

where $z^{(r)}$ and $z^{(g)}$ denote the Inception features of real and generated images, and $T$ controls the frequency scale at which the two distributions are compared. As shown in Table 2, JumpSampler consistently yields lower ECS than the baseline across multiple frequency scales for both raw and centered features. Notably, the improvement persists after subtracting the coordinate-wise mean from each distribution, suggesting that the gain is not merely due to correcting a global mean shift, but also reflects improved agreement in the higher-order structure of the Inception feature distribution.

**Text-to-image generation.** We further apply JumpSampler to SD3-M (Rombach et al., 2022) and FLUX.1 (Labs et al., 2025) and evaluate them on DPG-Bench (Hu et al., 2024). As shown in Table 3, the proposed method improves the overall prompt-alignment score for both models, with gains across several semantic categories. Although some individual categories exhibit minor trade-offs, the consistent gains in the overall score suggest that JumpSampler is not restricted to class-conditional generation and can be readily transferred to modern text-to-image models and efficient low-step sampling regimes. More results on text-to-image generation are provided in Appendix C.2.

Table 3: **DPG-Bench results on text-to-image generation.** We compare the baseline sampler with Ours on SD3-M and FLUX.1 across different compositional categories.

| Model (Default $\lambda = 1.0$) | Global↑ | Entity↑ | Attribute↑ | Relation↑ | Other↑ | **Overall↑** |
|---|---|---|---|---|---|---|
| SD3-M | 87.77 | 91.03 | 88.73 | 91.81 | 89.37 | 84.50 |
| SD3-M + Ours ($\beta = 2.8$) | **89.23** | **92.09** | **88.90** | 90.10 | 88.76 | **85.58** |
| FLUX.1 | 81.98 | 81.71 | 83.60 | 83.28 | 85.79 | 75.82 |
| FLUX.1 + Ours ($\beta = 2.0$) | 81.96 | **83.84** | **84.18** | **85.50** | 82.44 | **76.37** |

**Ablation on Jump Distribution** We further investigate the effect of the jump distribution on the resulting sample distribution. As shown in Table 4, among the tested distributions, the Student-$t$ distribution with $\nu = 3$ consistently achieves the lowest centered ECS across all evaluation scales. These results motivate our use of $\nu = 3$ as the default setting in the main experiments. More ablation results see Appendix C.3 and C.4.

Table 4: **Ablation of jump distributions on ImageNet-256 with SiT-XL/2.** We compare Gaussian jumps with Student-$t$ jumps of different degrees of freedom $\nu$.

| | Raw ECS ($\times 10^{-3}$) ↓ | | | | | Centered ECS ($\times 10^{-3}$) ↓ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $T$ | Baseline | Gaussian | $\nu = 2$ | $\nu = 3$ | $\nu = 4$ | Baseline | Gaussian | $\nu = 2$ | $\nu = 3$ | $\nu = 4$ |
| 0.02 | 5.282 | 5.280 | 5.279 | 5.278 | 5.279 | 0.0448 | 0.0436 | 0.0436 | **0.0435** | 0.0436 |
| 0.05 | 5.283 | 5.280 | 5.280 | 5.279 | 5.280 | 0.1120 | 0.1090 | 0.1090 | **0.1088** | 0.1091 |
| 0.10 | 5.286 | 5.283 | 5.282 | 5.282 | 5.282 | 0.2241 | 0.2181 | 0.2182 | **0.2177** | 0.2182 |
| 0.20 | 5.292 | 5.287 | 5.286 | 5.286 | 5.287 | 0.4477 | 0.4361 | 0.4361 | **0.4352** | 0.4361 |
| 0.50 | 5.281 | 5.273 | 5.273 | 5.273 | 5.273 | 1.0940 | 1.0698 | 1.0698 | **1.0677** | 1.0699 |
| 0.70 | 5.239 | 5.230 | 5.230 | 5.230 | 5.230 | 1.4845 | 1.4556 | 1.4556 | **1.4531** | 1.4558 |
| 1.00 | 5.129 | 5.118 | 5.117 | 5.118 | 5.118 | 1.9912 | 1.9584 | 1.9584 | **1.9551** | 1.9586 |
| 2.00 | 4.511 | 4.495 | 4.495 | 4.496 | 4.495 | 3.0449 | 3.0010 | 3.0010 | **2.9955** | 3.0014 |
| 3.00 | 3.783 | 3.774 | 3.773 | 3.776 | 3.774 | 3.4083 | 3.3650 | 3.3649 | **3.3604** | 3.3653 |

## 6 Conclusion

We introduced **JumpSampler**, a training-free sampling strategy that augments pretrained diffusion and flow models with sparse heavy-tailed jumps. Our theoretical analysis shows that the induced perturbation can be controlled while improving tail behavior, and experiments demonstrate consistent gains across class-conditional and text-to-image generation without additional neural network evaluations. These results highlight sparse nonlocal transitions as a simple and effective extension to conventional generative sampling dynamics.

**LLM Statement.** We use LLM to polish the writing, such as correcting grammar and other errors.

## REFERENCES


Junyeong Ahn. FD-DINOv2: FD Score via DINOv2. https://github.com/justin4ai/FD-DINOv2, June 2024. Version 0.1.0.

Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

David Applebaum. *Lévy processes and stochastic calculus*, volume 2. Cambridge university press Cambridge, 2009.

Ole E Barndorff-Nielsen and Neil Shephard. Econometric analysis of realized volatility and its use in estimating stochastic volatility models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 64(2):253–280, 2002.

Aymene Mohammed Bouayed, Samuel Deslauriers-Gauthier, Adrian Iaccovelli, and David Naccache. Conditional-$t^3$ VAE: Equitable latent space allocation for fair generation. *arXiv preprint arXiv:2509.02154*, 2025.

Stéphane Boucheron, Gábor Lugosi, and Olivier Bousquet. Concentration inequalities. In *Summer school on machine learning*, pp. 208–240. Springer, 2003.

Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

Rama Cont and Peter Tankov. *Financial modelling with jump processes*. CRC press, 2003.

Danilo de Oliveira, Julius Richter, Tal Peer, and Timo Gerkmann. Lipdiffuser: Lip-to-speech generation with conditional diffusion models. *arXiv preprint arXiv:2505.11391*, 2025.

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*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021. URL https://api.semanticscholar.org/CorpusID:234357997.

Luyao Fan, Jiayang Zou, Jiayang Gao, and Jia Wang. Differential properties of information in jump-diffusion channels. In *2025 IEEE International Symposium on Information Theory (ISIT)*, pp. 1–6. IEEE, 2025.

Tiziano Fassina, Gabriel Cardoso, Sylvan Le Corff, and Thomas Romary. Initialization-aware score-based diffusion sampling. *arXiv preprint arXiv:2603.00772*, 2026.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Floyd B Hanson. *Applied stochastic processes and control for jump-diffusions: modeling, analysis and computation*. SIAM, 2007.

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.

Jonathan Ho. Classifier-free diffusion guidance. *ArXiv*, abs/2207.12598, 2022. URL https://api.semanticscholar.org/CorpusID:249145348.

Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020. URL https://api.semanticscholar.org/CorpusID:219955663.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment, 2024.

Jisung Hwang, Jaihoon Kim, and Minhyuk Sung. Moment- and power-spectrum-based gaussianity regularization for text-to-image models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 18235–18264. Curran Associates, Inc., 2025. doi: 10.52202/085713-0614. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/1a67cdd6a9566ebe6e173a543687415d-Paper-Conference.pdf.

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.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proc. CVPR*, 2024.

Juno Kim, Jaehyuk Kwon, Mincheol Cho, Hyunjong Lee, and Joong-Ho Won. $t^3$-variational autoencoder: Learning heavy-tailed data with student's t and power divergence. In *International Conference on Learning Representations*, volume 2024, pp. 19580–19606, 2024.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ArXiv*, abs/2210.02747, 2022. URL https://api.semanticscholar.org/CorpusID:252734897.

Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *ArXiv*, abs/2209.14577, 2022. URL https://api.semanticscholar.org/CorpusID:252596014.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *ArXiv*, abs/2209.03003, 2022. URL https://api.semanticscholar.org/CorpusID:252111177.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Dilip B Madan, Peter P Carr, and Eric C Chang. The variance gamma process and option pricing. *Review of finance*, 2(1):79–105, 1998.

Robert C Merton. Option pricing when underlying stock returns are discontinuous. *Journal of financial economics*, 3(1-2):125–144, 1976.

Chrysostomos L Nikias and Min Shao. *Signal processing with alpha-stable distributions and applications*. Wiley-Interscience, 1995.

Kushagra Pandey, Jaideep Pathak, Yilun Xu, Stephan Mandt, Michael Pritchard, Arash Vahdat, and Morteza Mardani. Heavy-tailed diffusion models. *arXiv preprint arXiv:2410.14171*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

Lucas Pinheiro Cinelli, Matheus Araújo Marins, Eduardo Antúnio Barros da Silva, and Sérgio Lima Netto. Variational autoencoder. In *Variational methods for machine learning with applications to deep networks*, pp. 111–149. Springer, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Daniel Ruderman and William Bialek. Statistics of natural images: Scaling in the woods. *Advances in neural information processing systems*, 6, 1993.

Daniel L Ruderman. Origins of scaling in natural images. *Vision research*, 37(23):3385–3398, 1997.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Dario Shariatian, Umut Simsekli, and Alain Oliviero Durmus. Denoising levy probabilistic models. In *International Conference on Learning Representations*, volume 2025, pp. 96991–97024, 2025.

Jascha Narain Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *ArXiv*, abs/1503.03585, 2015. URL https://api.semanticscholar.org/CorpusID:14888175.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ArXiv*, abs/2010.02502, 2020a. URL https://api.semanticscholar.org/CorpusID:222140788.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Neural Information Processing Systems*, 2019. URL https://api.semanticscholar.org/CorpusID:196470871.

Yang Song, Jascha Narain Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ArXiv*, abs/2011.13456, 2020b. URL https://api.semanticscholar.org/CorpusID:227209335.

Stamatis Stamatelopoulos and Themistoklis P. Sapsis. Can diffusion models capture extreme event statistics? *Computer Methods in Applied Mechanics and Engineering*, 435:117589, 2025. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2024.117589. URL https://www.sciencedirect.com/science/article/pii/S0045782524008430.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Edric Tam and Barbara E Engelhardt. Assessing the distributional fidelity of synthetic chest x-rays using the embedded characteristic score. *arXiv preprint arXiv:2501.00744*, 2025.

Sangmin Woo, Byeongjun Park, Hyojun Go, Jin-Young Kim, and Changick Kim. Harmonyview: Harmonizing consistency and diversity in one-image-to-3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10574–10584, 2024.

Eun Bi Yoon, Keehun Park, Sungwoong Kim, and Sungbin Lim. Score-based generative models with lévy processes. *Advances in neural information processing systems*, 36:40694–40707, 2023.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

Tianyi Zheng, Cong Geng, Peng-Tao Jiang, Ben Wan, Hao Zhang, Jinwei Chen, Jia Wang, and Bo Li. Non-uniform timestep sampling: Towards faster diffusion model training. In *ACM Multimedia 2024*, 2024. URL https://openreview.net/forum?id=NQPJYEyiiM.

# APPENDIX

CONTENTS

**A Extended Related Work** **14**

A.1 Heavy-Tailed Generative Modeling. . . . . 14

A.2 Heavy-Tailed and Jump Modeling in Finance and Information Theory. 14

**B Proofs of Theorems** **15**

B.1 Proof of Proposition 1 . . . . . 15

B.2 Proof of Theorem 1 . . . . . 15

B.3 Propagation of Jump Perturbations . . . . . 18

B.4 Jump-Count Mixture and Student's $t$ Tail . . . . . 20

B.5 Proof of Theorem 2 . . . . . 20

B.6 Direct proof of Theorem 3 through the no-jump event . . . . . 21

B.7 Perturbation proof of Theorem 3 through the nonlocal Fokker–Planck equation . . . . . 22

B.8 Proof of Theorem 4 . . . . . 24

B.9 Proof of the Corollary 1 . . . . . 25

**C Additional Experiments** **26**

C.1 More Results of Toy Example . . . . . 26

C.2 Results on Geneval Benchmark . . . . . 27

C.3 Robustness across random seeds . . . . . 28

C.4 Sensitivity to jump intensity and temporal scheduling . . . . . 28

C.5 Generalization to Pixel-Space Second-Order Sampling . . . . . 28

C.6 Additional Visual Results . . . . . 29

## APPENDIX OVERVIEW

This appendix provides additional details and supplementary results to support the main paper. In Section A, we review more related work to place our work in a broader context. Section B presents the detailed proofs of the theoretical results introduced in the manuscript. Finally, Section C reports additional experimental results and visualizations.

## A EXTENDED RELATED WORK

### A.1 HEAVY-TAILED GENERATIVE MODELING.

Heavy-tailed and scale-free phenomena have long been observed in empirical data, although establishing a genuine power law requires careful statistical testing rather than visual inspection alone (Clauset et al., 2009). Natural-image statistics also exhibit pronounced departures from Gaussian behavior: Ruderman & Bialek (1993) reported approximately scale-invariant statistics of natural-image contrast, while Ruderman (1997) related the scaling behavior of natural images to a power-law distribution of object sizes. These observations do not imply that the full image distribution itself follows a power law, but they motivate generative models that can accommodate non-Gaussian and low-density structure.

Several recent works explicitly incorporate heavy-tailed distributions into generative modeling. Yoon et al. (2023) proposed the Lévy–Itô Model (LIM), replacing Brownian motion in score-based generative models with isotropic $\alpha$-stable Lévy processes and deriving the corresponding reverse-time SDE and fractional denoising score matching objective. Shariatian et al. (2025) instead extended the DDPM formulation directly by replacing Gaussian perturbations with $\alpha$-stable noise, resulting in Denoising Lévy Probabilistic Models (DLPMs), which improve tail coverage and generation on imbalanced data while requiring fewer backward steps in their experiments. Pandey et al. (2024) considered multivariate Student's $t$ distributions and developed heavy-tailed analogues of diffusion and flow models, including $t$-EDM and $t$-Flow, with tailored perturbation kernels and training objectives for modeling rare and extreme events. Heavy-tailed modeling has also been explored in latent-variable models. $t^3$-VAE (Kim et al., 2024) replaces Gaussian components of the VAE with Student's $t$ distributions and adopts a power-divergence-based objective to mitigate Gaussian over-regularization, while Conditional-$t^3$ VAE (Bouayed et al., 2025) further introduces class-conditional Student's $t$ priors and balanced latent mixtures to improve generation under severe class imbalance.

Complementary works study how departures from Gaussian assumptions affect generative inference without necessarily replacing the complete diffusion process. Hwang et al. (2025) introduce moment- and power-spectrum-based regularization that explicitly encourages latent samples to remain Gaussian during test-time optimization for text-to-image models. Fassina et al. (2026) analyze the role of reverse-process initialization and show that, for heavy-tailed targets, the mismatch between a light-tailed Gaussian initialization and the noised target distribution can impair tail reconstruction; they mitigate this effect by learning an initialization closer to the intermediate noised distribution while retaining the standard Gaussian noising process.

These approaches establish the relevance of heavy-tailed modeling at the noise-process, latent-prior, and initialization levels. However, they generally modify the generative formulation, training objective, latent model, or reverse-process initialization. In contrast, JumpSampler leaves the pretrained diffusion or flow model unchanged and introduces sparse heavy-tailed jumps solely at inference time. It therefore provides a complementary, sampler-side mechanism for introducing nonlocal heavy-tailed behavior without retraining the underlying generative model.

### A.2 HEAVY-TAILED AND JUMP MODELING IN FINANCE AND INFORMATION THEORY.

Heavy-tailed and jump processes have a long history in stochastic modeling beyond generative models. In mathematical finance, Merton (1976) extended continuous-path stock-price models by combining diffusion dynamics with a jump component, allowing the model to represent discontinuous price movements that cannot be captured by Brownian motion alone. Madan et al. (1998) developed the Variance Gamma process by evaluating Brownian motion with drift at a random time governed by a Gamma process, yielding a flexible non-Gaussian model with explicit control over skewness

and kurtosis. These models illustrate a general stochastic-modeling principle in which continuous Gaussian dynamics are augmented or replaced by non-Gaussian mechanisms to better represent atypical or abrupt fluctuations.

Related ideas also arise in information theory and signal processing. Nikias & Shao (1995) systematically studied $\alpha$-stable distributions and processes as non-Gaussian models for impulsive signals and noise, where rare large-amplitude disturbances are poorly described by conventional Gaussian assumptions. More recently, Fan et al. (2025) formulated communication channels through general jump-diffusion processes and studied the evolution of entropy and mutual information under such dynamics. They extended classical differential identities, including de Bruijn's identity and the I–MMSE relation, beyond purely diffusive channels.

These developments provide a broader stochastic perspective on our approach: when local Gaussian fluctuations alone are insufficient to describe non-Gaussian or impulsive behavior, jump or heavy-tailed components provide a natural mechanism for introducing occasional large transitions. JumpSampler adopts this principle in generative sampling, but differs in that the jump mechanism is introduced solely at inference time on top of a pretrained diffusion or flow sampler, without altering its training objective or underlying model.

# B Proofs of Theorems

## B.1 Proof of Proposition 1

*Proof.* Consider the one-dimensional construction

$$X \sim \mathrm{StudentT}(\nu), \qquad q_\phi(z \mid X = x) = \mathcal{N}(\varepsilon x, 1), \tag{25}$$

where $\nu > 2$ and $\varepsilon > 0$. For each $x$,

$$D_{\mathrm{KL}}\left(\mathcal{N}(\varepsilon x, 1) \,\|\, \mathcal{N}(0,1)\right) = \frac{\varepsilon^2 x^2}{2}. \tag{26}$$

Since $\mathbb{E}[X^2] = \nu/(\nu-2)$, averaging over $X$ gives

$$\mathbb{E}_X\left[D_{\mathrm{KL}}\left(q_\phi(z \mid X) \,\|\, \mathcal{N}(0,1)\right)\right] = \frac{\varepsilon^2}{2}\frac{\nu}{\nu-2}, \tag{27}$$

which can be made arbitrarily small by choosing $\varepsilon > 0$ sufficiently small.

On the other hand, a sample from the aggregated latent distribution can be written as

$$Z = \varepsilon X + \xi, \qquad \xi \sim \mathcal{N}(0,1), \tag{28}$$

with $X$ and $\xi$ independent. For every fixed $\varepsilon > 0$, the Student's $t$ component has polynomial tails, whereas the Gaussian component has exponentially decaying tails. Consequently,

$$\mathbb{P}(|Z| > r) \asymp r^{-\nu}, \qquad r \to \infty. \tag{29}$$

Thus, the aggregated latent distribution remains heavy-tailed even when the average Gaussian KL regularization is arbitrarily small. □

## B.2 Proof of Theorem 1

**Lemma 1** (Gaussian Concentration Inequality, Boucheron et al. (2003))**.** *Let $X = (X_1, X_2, \cdots, X_n)$ be a vector of $n$ independent standard normal random variables. Let $f : \mathbb{R}^n \to \mathbb{R}$ denote an $L$-Lipschitz function. Then, for all $t > 0$,*

$$\mathbb{P}\left(f(X) - \mathbb{E}f(X) \geq t\right) \leq \exp\left(-\frac{t^2}{2L^2}\right) \tag{30}$$

*Proof.* We now show that a finite composition of the updates in Equation 12, driven only by Gaussian random variables, has a sub-Gaussian terminal tail. Define the joint standard Gaussian vector

$$\Xi = (\xi_{\mathrm{init}}, \xi_K, \ldots, \xi_1)\,.$$

The complete base trajectory can be represented as a deterministic function of $\Xi$:

$$\bar{X}_0 = \mathcal{F}(\Xi). \tag{31}$$

For a perturbation introduced after the $k$-th base update, define the forward amplification factor

$$A_k^+ := \prod_{j=1}^{k-1} L_j^+, \qquad A_1^+ = 1, \tag{32}$$

and define the amplification factor from initialization to the terminal state as

$$A_{\text{init}}^+ := \prod_{j=1}^{K} L_j^+. \tag{33}$$

Finally, let

$$\Gamma^2 := \left(A_{\text{init}}^+ g_{\text{init}}\right)^2 + \sum_{k=1}^{K} \left(A_k^+ g_k\right)^2. \tag{34}$$

We first establish the Lipschitz dependence of the terminal state on all Gaussian inputs. Consider two joint Gaussian inputs

$$\Xi = (\xi_{\text{init}}, \xi_K, \ldots, \xi_1), \qquad \widetilde{\Xi} = \left(\widetilde{\xi}_{\text{init}}, \widetilde{\xi}_K, \ldots, \widetilde{\xi}_1\right),$$

and let $\{\bar{X}_k\}_{k=0}^K$ and $\{\widetilde{X}_k\}_{k=0}^K$ denote the corresponding trajectories. Define

$$\Delta_k := \left\| \bar{X}_k - \widetilde{X}_k \right\|_2.$$

From the initialization rule,

$$\bar{X}_K - \widetilde{X}_K = g_{\text{init}} \left(\xi_{\text{init}} - \widetilde{\xi}_{\text{init}}\right),$$

because the deterministic mean $m_K$ cancels. Therefore,

$$\Delta_K = |g_{\text{init}}| \left\| \xi_{\text{init}} - \widetilde{\xi}_{\text{init}} \right\|_2.$$

For the $k$-th update, we have

$$\bar{X}_{k-1} = \bar{X}_k + h_k F_k(\bar{X}_k) + g_k \xi_k + d_k$$

and

$$\widetilde{X}_{k-1} = \widetilde{X}_k + h_k F_k(\widetilde{X}_k) + g_k \widetilde{\xi}_k + d_k.$$

Subtracting the two identities gives

$$\bar{X}_{k-1} - \widetilde{X}_{k-1} = \left(\bar{X}_k - \widetilde{X}_k\right) + h_k \left(F_k(\bar{X}_k) - F_k(\widetilde{X}_k)\right) + g_k \left(\xi_k - \widetilde{\xi}_k\right).$$

Using the triangle inequality and the Lipschitz property of $F_k$, we obtain

$$\begin{aligned} \Delta_{k-1} &\le \Delta_k + h_k M_k \Delta_k + |g_k| \left\| \xi_k - \widetilde{\xi}_k \right\|_2 \\ &= L_k^+ \Delta_k + |g_k| \left\| \xi_k - \widetilde{\xi}_k \right\|_2. \end{aligned}$$

Thus, for every $k = 1, \ldots, K$,

$$\Delta_{k-1} \le L_k^+ \Delta_k + |g_k| \left\| \xi_k - \widetilde{\xi}_k \right\|_2.$$

We now iterate this recursion. At the first backward update,

$$\Delta_0 \le L_1^+ \Delta_1 + |g_1| \left\| \xi_1 - \widetilde{\xi}_1 \right\|_2.$$

Applying the same inequality to $\Delta_1$ gives

$$\begin{aligned}\Delta_0 &\le L_1^+ \left[ L_2^+ \Delta_2 + |g_2| \left\| \xi_2 - \widetilde{\xi}_2 \right\|_2 \right] + |g_1| \left\| \xi_1 - \widetilde{\xi}_1 \right\|_2 \\ &= L_1^+ L_2^+ \Delta_2 + L_1^+ |g_2| \left\| \xi_2 - \widetilde{\xi}_2 \right\|_2 + |g_1| \left\| \xi_1 - \widetilde{\xi}_1 \right\|_2 .\end{aligned}$$

Continuing inductively yields

$$\Delta_0 \le \left( \prod_{j=1}^{K} L_j^+ \right) \Delta_K + \sum_{k=1}^{K} \left( \prod_{j=1}^{k-1} L_j^+ \right) |g_k| \left\| \xi_k - \widetilde{\xi}_k \right\|_2 .$$

Using the definitions of $A_{\text{init}}^+$ and $A_k^+$, together with the preceding expression for $\Delta_K$, we obtain

$$\left\| \bar{X}_0 - \widetilde{X}_0 \right\|_2 \le A_{\text{init}}^+ |g_{\text{init}}| \left\| \xi_{\text{init}} - \widetilde{\xi}_{\text{init}} \right\|_2 + \sum_{k=1}^{K} A_k^+ |g_k| \left\| \xi_k - \widetilde{\xi}_k \right\|_2 .$$

Applying the Cauchy–Schwarz inequality to this finite sum gives

$$\left\| \bar{X}_0 - \widetilde{X}_0 \right\|_2 \le \left( (A_{\text{init}}^+ |g_{\text{init}}|)^2 + \sum_{k=1}^{K} (A_k^+ |g_k|)^2 \right)^{1/2} \left( \left\| \xi_{\text{init}} - \widetilde{\xi}_{\text{init}} \right\|_2^2 + \sum_{k=1}^{K} \left\| \xi_k - \widetilde{\xi}_k \right\|_2^2 \right)^{1/2} .$$

By the definition of the Euclidean norm on the product space $\mathbb{R}^{d(K+1)}$,

$$\left( \left\| \xi_{\text{init}} - \widetilde{\xi}_{\text{init}} \right\|_2^2 + \sum_{k=1}^{K} \left\| \xi_k - \widetilde{\xi}_k \right\|_2^2 \right)^{1/2} = \left\| \Xi - \widetilde{\Xi} \right\|_2 .$$

Moreover,

$$\left( A_{\text{init}}^+ g_{\text{init}} \right)^2 + \sum_{k=1}^{K} \left( A_k^+ g_k \right)^2 = \Gamma^2 .$$

Consequently,

$$\left\| \mathcal{F}(\Xi) - \mathcal{F}(\widetilde{\Xi}) \right\|_2 \le \Gamma \left\| \Xi - \widetilde{\Xi} \right\|_2 ,$$

so $\mathcal{F}$ is $\Gamma$-Lipschitz.

Next, define the scalar function

$$\varphi(\Xi) := \left\| \mathcal{F}(\Xi) \right\|_2 .$$

Since the Euclidean norm is 1-Lipschitz,

$$\begin{aligned}\left| \varphi(\Xi) - \varphi(\widetilde{\Xi}) \right| &= \left| \left\| \mathcal{F}(\Xi) \right\|_2 - \left\| \mathcal{F}(\widetilde{\Xi}) \right\|_2 \right| \\ &\le \left\| \mathcal{F}(\Xi) - \mathcal{F}(\widetilde{\Xi}) \right\|_2 \\ &\le \Gamma \left\| \Xi - \widetilde{\Xi} \right\|_2 .\end{aligned}$$

Thus, $\varphi$ is also $\Gamma$-Lipschitz.

Because all Gaussian variables are mutually independent, the concatenated vector $\Xi$ is a standard Gaussian random vector in $\mathbb{R}^{d(K+1)}$. By Lemma 1, for every $r > 0$,

$$\mathbb{P}\left( \varphi(\Xi) > \mathbb{E}[\varphi(\Xi)] + r \right) \le \exp\left( -\frac{r^2}{2\Gamma^2} \right) .$$

Since $\varphi(\Xi) = \|X_0^0\|_2$, this proves

$$\mathbb{P}\left( \left\| X_0^0 \right\|_2 > \mathbb{E}\left[ \left\| X_0^0 \right\|_2 \right] + r \right) \le \exp\left( -\frac{r^2}{2\Gamma^2} \right) .$$

It remains to show that this concentration bound implies a tail decay faster than every polynomial. Let

$$m_0 := \mathbb{E}\left[\left\|X_0^0\right\|_2\right].$$

For every $R > m_0$, applying the preceding concentration inequality with $r = R - m_0$ gives

$$\mathbb{P}\left(\left\|X_0^0\right\|_2 > R\right) \le \exp\left(-\frac{(R-m_0)^2}{2\Gamma^2}\right).$$

If $R \ge 2m_0$, then $R - m_0 \ge R/2$, and hence

$$\mathbb{P}\left(\left\|X_0^0\right\|_2 > R\right) \le \exp\left(-\frac{R^2}{8\Gamma^2}\right).$$

Fix any finite $\nu > 0$. Multiplying the preceding inequality by $R^\nu$ yields

$$R^\nu \mathbb{P}\left(\left\|X_0^0\right\|_2 > R\right) \le R^\nu \exp\left(-\frac{R^2}{8\Gamma^2}\right).$$

Thus,

$$\mathbb{P}\left(\left\|X_0^0\right\|_2 > R\right) = o\left(R^{-\nu}\right), \quad R \to \infty.$$

Finally, suppose that $\Gamma = 0$. Since all amplification factors are strictly positive, the identity

$$\Gamma^2 = \left(A_{\text{init}}^+ g_{\text{init}}\right)^2 + \sum_{k=1}^{K} \left(A_k^+ g_k\right)^2 = 0$$

implies

$$g_{\text{init}} = g_1 = \cdots = g_K = 0.$$

Hence the initialization and all subsequent solver updates are deterministic, so $\bar{X}_0$ is deterministic. It follows that

$$\mathbb{P}\left(\left\|X_0^0\right\|_2 > R\right) = 0$$

for every sufficiently large $R$. Thus, the conclusion

$$\mathbb{P}\left(\left\|X_0^0\right\|_2 > R\right) = o\left(R^{-\nu}\right)$$

also holds when $\Gamma = 0$. □

### B.3 Propagation of Jump Perturbations

**Lemma 2** (Single-step bi-Lipschitz property)**.** *Under Assumption 1 and 2, for any $x, y \in \mathbb{R}^d$ and the same realization of $\xi_k$,*

$$L_k^- \|x - y\|_2 \le \|\Psi_k(x;\xi_k) - \Psi_k(y;\xi_k)\|_2 \le L_k^+ \|x - y\|_2. \tag{35}$$

*Proof.* For the same $\xi_k$, the additive Gaussian and deterministic terms cancel, and hence

$$\Psi_k(x;\xi_k) - \Psi_k(y;\xi_k) = (x - y) + h_k\big(F_k(x) - F_k(y)\big).$$

The upper bound follows from the triangle inequality:

$$\|\Psi_k(x;\xi_k) - \Psi_k(y;\xi_k)\|_2 \le (1 + h_k M_k)\|x - y\|_2 = L_k^+ \|x - y\|_2.$$

Similarly, the reverse triangle inequality gives

$$\|\Psi_k(x;\xi_k) - \Psi_k(y;\xi_k)\|_2 \ge (1 - h_k M_k)\|x - y\|_2 = L_k^- \|x - y\|_2.$$

□

**Remark 1.** *Lemma 2 states that, under a synchronous coupling of the base randomness, one solver step can neither amplify a state discrepancy without bound nor collapse an arbitrarily large discrepancy to zero.*

JumpSampler uses the same initialization and Gaussian solver variables as the base sampler and adds a jump after the $k$-th base update:

$$X_{k-1} = \Psi_k(X_k; \xi_k) + \delta_k, \tag{36}$$

$$\delta_k = B_k a_k \rho_k u_k, \qquad a_k := \left(\frac{t_k}{T}\right)^{\beta}, \tag{37}$$

where $B_k \sim \mathrm{Bernoulli}(p_k)$, $\rho_k \sim |\mathrm{StudentT}(\nu)|$, and $\|u_k\|_2 = 1$. All jump variables are independent of the Gaussian variables of the base sampler. Let

$$e_k := X_k - X_k^0$$

denote the discrepancy under this synchronous coupling.

**Proposition 2** (Pathwise upper bound for multiple jumps)**.** *Under Assumption 1, for every realization of the jump variables,*

$$\|e_0\|_2 \le \sum_{k=1}^{K} A_k^+ \|\delta_k\|_2 . \tag{38}$$

*In particular, if exactly $n$ jumps occur at steps $k_1, \dots, k_n$, then*

$$\left\| X_0^{(n)} - X_0^0 \right\|_2 \le A_+ \sum_{i=1}^{n} \|Z_i\|_2 , \tag{39}$$

*where $Z_i$ denotes the corresponding nonzero jump vector and*

$$A_+ := \max_{1 \le k \le K} A_k^+ < \infty. \tag{40}$$

*Proof.* By equation 36 and the upper bound in Lemma 2,

$$\|e_{k-1}\|_2 \le L_k^+ \|e_k\|_2 + \|\delta_k\|_2 .$$

Since $e_K = 0$, iterating this inequality gives

$$\|e_0\|_2 \le \sum_{k=1}^{K} \left( \prod_{j=1}^{k-1} L_j^+ \right) \|\delta_k\|_2 = \sum_{k=1}^{K} A_k^+ \|\delta_k\|_2 .$$

The second statement follows because only the $n$ activated jump vectors are nonzero. □

To obtain a matching tail lower bound, we also need to show that a single jump cannot be completely erased by the remaining base-solver updates. Define

$$A_k^- := \prod_{j=1}^{k-1} L_j^- , \qquad A_1^- = 1. \tag{41}$$

**Proposition 3** (Non-collapse of a single jump)**.** *Suppose that the trajectory contains exactly one nonzero jump $Z$ at step $k$ and we denote the terminal sample as $X_0^{(1,k)}$. Then under Assumption 2,*

$$\left\| X_0^{(1,k)} - X_0^0 \right\|_2 \ge A_k^- \|Z\|_2 . \tag{42}$$

*In particular, $A_k^- > 0$ because the number of solver steps is finite and $L_j^- > 0$ for every $j$.*

*Proof.* Immediately after the jump is added,

$$\left\| X_{k-1}^{(1,k)} - X_{k-1}^0 \right\|_2 = \|Z\|_2 .$$

The two trajectories subsequently use the same Gaussian variables. Repeated application of the lower bound in Lemma 2 therefore yields

$$\left\| X_0^{(1,k)} - X_0^0 \right\|_2 \ge \left( \prod_{j=1}^{k-1} L_j^- \right) \|Z\|_2 = A_k^- \|Z\|_2 .$$

□

Propositions 2 and 3 derive the two propagation properties needed for the tail analysis directly from the bi-Lipschitz regularity of the base solver: jump perturbations have a finite upper amplification factor, while a single jump retains a strictly positive fraction of its magnitude at the terminal state.

### B.4 JUMP-COUNT MIXTURE AND STUDENT'S $t$ TAIL

Let

$$N_K := \sum_{k=1}^{K} B_k \tag{43}$$

denote the total number of activated jumps and define

$$w_n := \mathbb{P}(N_K = n), \qquad 0 \le n \le K.$$

Let $Q^{(n)}$ denote the conditional terminal law of $X_0$ given $N_K = n$. The terminal law $Q^J$ of JumpSampler then admits the finite mixture decomposition

$$Q^J = \sum_{n=0}^{K} w_n Q^{(n)}, \qquad Q^{(0)} = Q^0, \tag{44}$$

where $Q^0$ is the terminal law of the base sampler. Accordingly,

$$\mathbb{P}\big(\|X_0^J\|_2 > r\big) = \sum_{n=0}^{K} w_n \mathbb{P}\Big(\Big\|X_0^{(n)}\Big\|_2 > r\Big). \tag{45}$$

This is a positive finite mixture of probability laws, so the tail contribution of the jump-active components cannot be removed by sign cancellation.

For

$$\rho \sim |\mathrm{StudentT}(\nu)|,$$

the density satisfies $f_\rho(r) \asymp r^{-(\nu+1)}$ as $r \to \infty$. Hence there exist constants $0 < c_\rho \le C_\rho < \infty$ and $r_\rho > 0$ such that, for every $r \ge r_\rho$,

$$c_\rho r^{-\nu} \le \mathbb{P}(\rho > r) \le C_\rho r^{-\nu}. \tag{46}$$

After increasing $C_\rho$ if necessary, we also have the global upper bound

$$\mathbb{P}(\rho > r) \le C_\rho (1 + r)^{-\nu}, \qquad r \ge 0. \tag{47}$$

### B.5 PROOF OF THEOREM 2

*Proof.* Define the event that the only activated jump occurs at step $k_*$:

$$\mathcal{E}_{k_*} := \{B_{k_*} = 1\} \cap \bigcap_{j \ne k_*} \{B_j = 0\}. \tag{48}$$

Its probability is

$$\pi_{k_*} := p_{k_*} \prod_{j \ne k_*} (1 - p_j) > 0. \tag{49}$$

We first establish the lower bound. On the exact-single-jump event $\mathcal{E}_{k_*}$, Proposition 3 gives

$$\big\|X_0^J - \bar{X}_0\big\|_2 \ge A_{k_*}^- a_{k_*} \rho_{k_*}.$$

There exists $M > 0$ such that

$$q_M := \mathbb{P}\big(\big\|\bar{X}_0\big\|_2 \le M\big) > 0. \tag{50}$$

By the reverse triangle inequality, on the event $\mathcal{E}_{k_*} \cap \{\|\bar{X}_0\|_2 \le M\}$,

$$\big\|X_0^J\big\|_2 \ge A_{k_*}^- a_{k_*} \rho_{k_*} - M.$$

The trigger variables, the jump radii, and the Gaussian variables of the base sampler are mutually independent. Therefore,

$$\mathbb{P}\big(\big\|X_0^J\big\|_2 > r\big) \ge \pi_{k_*} q_M \mathbb{P}\left(\rho_{k_*} > \frac{r + M}{A_{k_*}^- a_{k_*}}\right).$$

Using the lower bound in equation 46, we obtain

$$\mathbb{P}\big(\|X_0^J\|_2 > r\big) \geq c_J r^{-\nu}$$

for all sufficiently large $r$, where $c_J > 0$.

We next prove the upper bound. When $N_K = 0$, Theorem 1 directly implies that, for all sufficiently large $r$,

$$\mathbb{P}\Big(\Big\|X_0^{(0)}\Big\|_2 > r\Big) \leq C_0 r^{-\nu}$$

for some $C_0 < \infty$. Now fix $1 \leq n \leq K$. By Proposition 2, the triangle inequality, and $0 \leq a_k \leq 1$,

$$\Big\|X_0^{(n)}\Big\|_2 \leq \|X_0^0\|_2 + A_+ \sum_{i=1}^{n} \rho_i.$$

Consequently,

$$\mathbb{P}\Big(\Big\|X_0^{(n)}\Big\|_2 > r\Big) \leq \mathbb{P}\Big(\|X_0^0\|_2 > \frac{r}{2}\Big) + \mathbb{P}\left(\sum_{i=1}^{n} \rho_i > \frac{r}{2A_+}\right).$$

The first term is at most $C_0 r^{-\nu}$ for all sufficiently large $r$. For the second term, if

$$\sum_{i=1}^{n} \rho_i > \frac{r}{2A_+},$$

then at least one $i$ satisfies $\rho_i > r/(2A_+ n)$. The union bound and equation 47 therefore give

$$\begin{aligned}\mathbb{P}\left(\sum_{i=1}^{n} \rho_i > \frac{r}{2A_+}\right) &\leq n\,\mathbb{P}\Big(\rho_1 > \frac{r}{2A_+ n}\Big) \\ &\leq C_\rho (2A_+)^\nu n^{\nu+1} r^{-\nu}\end{aligned}$$

for all sufficiently large $r$. Hence, for every $0 \leq n \leq K$, there exists $C_n < \infty$ such that

$$\mathbb{P}\Big(\Big\|X_0^{(n)}\Big\|_2 > r\Big) \leq C_n r^{-\nu} \tag{51}$$

for all sufficiently large $r$. Substituting this bound into the finite mixture equation 45 yields

$$\mathbb{P}\big(\|X_0^J\|_2 > r\big) \leq \left(\sum_{n=0}^{K} w_n C_n\right) r^{-\nu}.$$

Because $K$ is finite, $C_J := \sum_{n=0}^{K} w_n C_n < \infty$, which proves the upper bound in equation 17. □

### B.6 Direct proof of Theorem 3 through the no-jump event

*Proof.* Define the cumulative jump intensity by

$$\Lambda_s := \int_0^s r(\eta)\,\mathrm{d}\eta, \quad \Lambda_T = \frac{\lambda}{\beta+1}. \tag{52}$$

Let $N_s$ denote the number of activated jumps over the sampling interval $[0, s]$. Since $N_s$ is a non-homogeneous Poisson counting process,

$$N_s \sim \mathrm{Poisson}(\Lambda_s). \tag{53}$$

Define the no-jump event

$$\mathcal{A}_0(s) := \{N_s = 0\}. \tag{54}$$

Its probability is

$$\pi_0(s) := \mathbb{P}\big(\mathcal{A}_0(s)\big) = \exp(-\Lambda_s). \tag{55}$$

On the event $\mathcal{A}_0(s)$, no jump occurs before time $s$. Since the base process and JumpSampler start from the same initial state and use the same Brownian motion, pathwise uniqueness implies

$$X_\eta^J = X_\eta^0, \qquad 0 \le \eta \le s, \tag{56}$$

on $\mathcal{A}_0(s)$.

Moreover, the event $\mathcal{A}_0(s)$ depends only on the jump process and is independent of the base-sampler randomness. Therefore, the conditional law of $X_s^J$ given $\mathcal{A}_0(s)$ is exactly $Q_s^0$:

$$\mathcal{L}\left(X_s^J \,\middle|\, \mathcal{A}_0(s)\right) = Q_s^0. \tag{57}$$

Let

$$Q_s^{\text{active}} := \mathcal{L}\left(X_s^J \,\middle|\, N_s \ge 1\right) \tag{58}$$

denote the terminal law conditioned on at least one activated jump. By the law of total probability,

$$Q_s^J = \pi_0(s) Q_s^0 + \big(1 - \pi_0(s)\big) Q_s^{\text{active}}. \tag{59}$$

Subtracting $Q_s^0$ from both sides yields

$$Q_s^J - Q_s^0 = \big(1 - \pi_0(s)\big)\left(Q_s^{\text{active}} - Q_s^0\right). \tag{60}$$

Using the homogeneity of total variation distance,

$$\mathrm{TV}\left(Q_s^J, Q_s^0\right) = \big(1 - \pi_0(s)\big)\,\mathrm{TV}\left(Q_s^{\text{active}}, Q_s^0\right). \tag{61}$$

Since the total variation distance between any two probability measures is at most one,

$$\begin{aligned} \mathrm{TV}\left(Q_s^J, Q_s^0\right) &\le 1 - \pi_0(s) \\ &= 1 - \exp(-\Lambda_s). \end{aligned} \tag{62}$$

Finally, the elementary inequality

$$1 - \exp(-x) \le x, \qquad x \ge 0, \tag{63}$$

gives

$$\mathrm{TV}\left(Q_s^J, Q_s^0\right) \le \Lambda_s. \tag{64}$$

□

**Remark 2.** *The bound in Theorem 3 depends only on the probability that at least one jump occurs. It does not depend on the magnitude or tail behavior of the jump marks. Consequently, even an unbounded heavy-tailed jump distribution produces a controlled total variation perturbation when the cumulative jump intensity $\Lambda_T$ is small.*

### B.7 Perturbation proof of Theorem 3 through the nonlocal Fokker–Planck equation

We next derive an alternative total variation bound by treating the jump operator as a nonlocal perturbation of the base Fokker–Planck evolution. This proof requires the marginal laws to admit probability densities.

Let $p_s^0$ and $p_s^J$ denote the densities of $X_s^0$ and $X_s^J$, respectively. The base density satisfies

$$\partial_s p_s^0 = \mathcal{L}_s^* p_s^0, \tag{65}$$

where $\mathcal{L}_s^*$ is the Fokker–Planck operator associated with the base drift and diffusion.

For state-independent additive compound Poisson jumps, the jump-augmented density satisfies

$$\partial_s p_s^J = \mathcal{L}_s^* p_s^J + \mathcal{J}_s^* p_s^J, \tag{66}$$

where the nonlocal jump operator is

$$\mathcal{J}_s^* f(x) := r_s \int_{\mathbb{R}^d} [f(x - z) - f(x)]\, \mu_s(\mathrm{d}z). \tag{67}$$

Here, $\mu_s$ denotes the probability distribution of a single jump vector at sampling time $s$.

We assume that the base dynamics generate a conservative Markov evolution family $U(s, \eta)$ satisfying

$$\|U(s, \eta) f\|_{L^1} \le \|f\|_{L^1}, \qquad 0 \le \eta \le s \le T. \tag{68}$$

**Lemma 3** ($L^1$ bound for the jump operator)**.** *For every $f \in L^1(\mathbb{R}^d)$,*

$$\|\mathcal{J}_s^* f\|_{L^1} \le 2r_s \|f\|_{L^1}. \tag{69}$$

*Proof.* By Equation equation 67 and Minkowski's inequality,

$$\|\mathcal{J}_s^* f\|_{L^1} \le r_s \int_{\mathbb{R}^d} \|f(\cdot - z) - f(\cdot)\|_{L^1} \, \mu_s(\mathrm{d}z). \tag{70}$$

Using the triangle inequality and the translation invariance of the $L^1$ norm,

$$\begin{aligned} \|f(\cdot - z) - f(\cdot)\|_{L^1} &\le \|f(\cdot - z)\|_{L^1} + \|f\|_{L^1} \\ &= 2\|f\|_{L^1}. \end{aligned} \tag{71}$$

Since $\mu_s$ is a probability measure,

$$\|\mathcal{J}_s^* f\|_{L^1} \le 2r_s \|f\|_{L^1}. \tag{72}$$

□

**Theorem 5** (Total variation stability via operator perturbation)**.** *Suppose that the base dynamics generate a conservative Markov evolution family satisfying Equation equation 68, and that $p_s^J$ is a probability density for every $s \in [0, T]$. Then*

$$\left\|p_s^J - p_s^0\right\|_{L^1} \le 2 \int_0^s r_\eta \, \mathrm{d}\eta = 2\Lambda_s. \tag{73}$$

*Consequently,*

$$\mathrm{TV}\left(Q_s^J, Q_s^0\right) \le \Lambda_s. \tag{74}$$

*Proof.* The mild solution of Equation equation 66 is

$$p_s^J = U(s, 0)p_0 + \int_0^s U(s, \eta)\mathcal{J}_\eta^* p_\eta^J \, \mathrm{d}\eta. \tag{75}$$

The base density satisfies

$$p_s^0 = U(s, 0)p_0. \tag{76}$$

Subtracting Equation equation 76 from Equation equation 75 gives

$$p_s^J - p_s^0 = \int_0^s U(s, \eta)\mathcal{J}_\eta^* p_\eta^J \, \mathrm{d}\eta. \tag{77}$$

Taking the $L^1$ norm and using the contraction property of the base evolution,

$$\begin{aligned} \left\|p_s^J - p_s^0\right\|_{L^1} &\le \int_0^s \left\|U(s, \eta)\mathcal{J}_\eta^* p_\eta^J\right\|_{L^1} \, \mathrm{d}\eta \\ &\le \int_0^s \left\|\mathcal{J}_\eta^* p_\eta^J\right\|_{L^1} \, \mathrm{d}\eta. \end{aligned} \tag{78}$$

By Lemma 3,

$$\left\|\mathcal{J}_\eta^* p_\eta^J\right\|_{L^1} \le 2r_\eta \left\|p_\eta^J\right\|_{L^1}. \tag{79}$$

Since $p_\eta^J$ is a probability density,

$$\left\|p_\eta^J\right\|_{L^1} = 1. \tag{80}$$

Therefore,

$$\left\|p_s^J - p_s^0\right\|_{L^1} \le 2 \int_0^s r_\eta \, \mathrm{d}\eta = 2\Lambda_s. \tag{81}$$

Using

$$\mathrm{TV}(P, Q) = \frac{1}{2}\|p - q\|_{L^1} \tag{82}$$

for probability measures with densities proves Equation equation 74. □

**Remark 3.** *The no-jump decomposition gives the sharper bound*

$$\mathrm{TV}\left(Q_s^J, Q_s^0\right) \le 1 - \exp(-\Lambda_s), \tag{83}$$

*whereas the operator perturbation argument gives*

$$\mathrm{TV}\left(Q_s^J, Q_s^0\right) \le \Lambda_s. \tag{84}$$

*Since $1 - \exp(-x) \le x$, the first bound is always at least as tight as the second. The purpose of the operator proof is not to improve the numerical constant, but to show that the nonlocal jump generator acts as an $L^1$-controlled perturbation of the base Fokker–Planck evolution.*

For the intensity schedule used by JumpSampler, let

$$r_s = \frac{\lambda}{T}\left(1 - \frac{s}{T}\right)^{\beta}. \tag{85}$$

Then

$$\Lambda_T = \int_0^T r_s \,\mathrm{d}s = \frac{\lambda}{\beta + 1}. \tag{86}$$

Consequently,

$$\begin{aligned}\mathrm{TV}\left(Q_T^J, Q_T^0\right) &\le 1 - \exp\left(-\frac{\lambda}{\beta+1}\right) \\ &\le \frac{\lambda}{\beta+1}.\end{aligned} \tag{87}$$

**Remark 4** (Discrete-time counterpart)**.** *For the finite-step implementation with independent activation variables $B_k \sim \mathrm{Bernoulli}(p_k)$, the same no-jump argument gives*

$$\mathrm{TV}\left(Q_{\mathrm{disc}}^J, Q_{\mathrm{disc}}^0\right) \le 1 - \prod_{k=1}^{K}(1 - p_k). \tag{88}$$

*When $p_k = r(t_k)h_k + o(h_k)$ and the inference grid is refined,*

$$\prod_{k=1}^{K}(1 - p_k) \longrightarrow \exp\left(-\int_0^T r(t)\,\mathrm{d}t\right). \tag{89}$$

*Thus, the continuous-time TV bound is the natural limiting counterpart of the finite-step Bernoulli sampler.*

### B.8 Proof of Theorem 4

*Proof.* By the definition of the quadratic Wasserstein distance,

$$W_2^2(Q^J, Q^0) = \inf_{\gamma \in \Pi(Q^J, Q^0)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|_2^2 \,\gamma(\mathrm{d}x, \mathrm{d}y),$$

where $\Pi(Q^J, Q^0)$ denotes the set of all couplings between $Q^J$ and $Q^0$. The synchronous construction $(X_0^J, X_0^0)$ is one admissible coupling. Therefore,

$$W_2^2(Q^J, Q^0) \le \mathbb{E}\left[\|X_0^J - X_0^0\|_2^2\right].$$

Define

$$e_{k-1} = X_{k-1}^J - \bar{X}_{k-1} = \Psi_k(X_k^J; \xi_k) - \Psi_k(\bar{X}_k; \xi_k) + \delta_k.$$

$$D_k := \Psi_k(X_k^J; \xi_k) - \Psi_k(\bar{X}_k; \xi_k).$$

Then

$$\|e_{k-1}\|_2^2 = \|D_k\|_2^2 + 2\langle D_k, \delta_k\rangle + \|\delta_k\|_2^2.$$

Since $D_k$ is independent of $\delta_k$ and $\mathbb{E}[\delta_k] = 0$, we have

$$\mathbb{E}\left[\langle D_k, \delta_k \rangle\right] = 0.$$

By the upper Lipschitz bound implied by Assumption 1,

$$\|D_k\|_2 \leq L_k^+ \|e_k\|_2.$$

Consequently,

$$\mathbb{E}\left[\|e_{k-1}\|_2^2\right] \leq (L_k^+)^2 \mathbb{E}\left[\|e_k\|_2^2\right] + \mathbb{E}\left[\|\delta_k\|_2^2\right]. \tag{90}$$

Since $e_K = 0$, iterating Equation 90 from $k = K$ down to $k = 1$ gives

$$\begin{aligned}\mathbb{E}\left[\|e_0\|_2^2\right] &\leq \sum_{k=1}^{K} \left(\prod_{j=1}^{k-1} L_j^+\right)^2 \mathbb{E}\left[\|\delta_k\|_2^2\right] \\ &= \sum_{k=1}^{K} (A_k^+)^2 \mathbb{E}\left[\|\delta_k\|_2^2\right].\end{aligned}$$

Using

$$\mathbb{E}\left[\|\delta_k\|_2^2\right] = p_k a_k^2 m_2$$

completes the proof. □

### B.9 Proof of the Corollary 1

*Proof.* Using Equation 32 and the inequality $1 + x \leq \exp(x)$,

$$\begin{aligned}A_k^+ &= \prod_{j=1}^{k-1} L_j^+ \\ &\leq \prod_{j=1}^{k-1} (1 + Ch_j) \\ &\leq \exp\left(C \sum_{j=1}^{k-1} h_j\right) \\ &\leq \exp(CT).\end{aligned}$$

Substituting this bound into Equation 22 gives

$$W_2^2(Q^J, Q^0) \leq \frac{\lambda m_2 \exp(2CT)}{T} \sum_{k=1}^{K} \left(\frac{t_k}{T}\right)^{3\beta} h_k.$$

Since $(t_k/T)^{3\beta} \leq 1$ and $\sum_{k=1}^{K} h_k = T$, we obtain

$$W_2^2(Q^J, Q^0) \leq \lambda m_2 \exp(2CT).$$

Finally, the Riemann-sum convergence

$$\begin{aligned}\frac{1}{T} \sum_{k=1}^{K} \left(\frac{t_k}{T}\right)^{3\beta} h_k &\longrightarrow \int_0^1 s^{3\beta} \, \mathrm{d}s \\ &= \frac{1}{3\beta + 1}\end{aligned}$$

proves Equation 23. □

# C Additional Experiments

## C.1 More Results of Toy Example

**Experimental setup.** We construct a one-dimensional asymmetric heavy-tailed target distribution from two truncated Pareto components. Specifically, with probability $0.3$, a sample is drawn from the positive component

$$X = R_+, \qquad R_+ \sim \mathrm{Pareto}(\alpha_+ = 2.0, x_{\min,+} = 1.5),$$

truncated at $x_{\max,+} = 80$, while with probability $0.7$, it is drawn from the negative component

$$X = -R_-, \qquad R_- \sim \mathrm{Pareto}(\alpha_- = 2.0, x_{\min,-} = 1.0),$$

truncated at $x_{\max,-} = 40$. This construction yields an asymmetric distribution with polynomially decaying tails on both sides while keeping the numerical range finite for stable training and evaluation.

We train a one-dimensional rectified-flow model to transport a standard Gaussian source $X_0 \sim \mathcal{N}(0, 1)$ to the target distribution. For independently sampled source–target pairs $(X_0, X_1)$, the interpolation is

$$X_t = (1 - t)X_0 + tX_1, \qquad t \sim \mathcal{U}(0, 1),$$

and the velocity network is trained with the standard flow-matching objective

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}\left[|v_\theta(X_t, t) - (X_1 - X_0)|^2\right].$$

The velocity field is parameterized by a four-layer MLP with hidden dimension $128$ and SiLU activations, together with a sinusoidal time embedding using eight frequencies. We train for $50{,}000$ iterations with batch size $2048$ using AdamW with learning rate $2 \times 10^{-4}$ and weight decay $10^{-4}$.

At inference time, the baseline model is sampled with a 70-step Euler solver. JumpSampler uses exactly the same pretrained velocity field and solver, and only augments each Euler update with the proposed stochastic jump operator. For this experiment, we set $\lambda = 1.0$, $\beta = 2.5$, $\nu = 3$, and the jump scale to $0.5$, with a maximum per-step activation probability of $p_{\max} = 0.08$. Since the purpose of this experiment is to test whether heavy-tailed perturbations can improve tail recovery without injecting prior knowledge of the target asymmetry, the jump sign is sampled symmetrically, with equal probability of a positive or negative displacement. We use $200{,}000$ independently generated samples for each distribution in the evaluation.

**Evaluation metrics.** In addition to the qualitative density and CCDF comparisons in Figure 2, we evaluate both global distribution fitting and tail-specific discrepancies. For overall fitting, we report the Kolmogorov–Smirnov (KS) distance, the one-dimensional Wasserstein-1 distance, the absolute error of the sample mean, and the relative error of the standard deviation. To more directly measure tail recovery, the two tails are evaluated separately. We compute the RMSE between the logarithmic complementary CDFs over the tail region, relative errors of extreme quantiles, errors in the probability mass beyond target-defined extreme quantile thresholds, and the error of the Hill tail-index estimate. The latter measures how accurately the generated samples recover the polynomial tail exponent of the target distribution.

**Quantitative results.** Table 5 shows that the improvement observed visually in Fig. 5 is consistently reflected by the quantitative metrics. At the distribution level, adding jumps reduces the KS distance from $0.0181$ to $0.0129$ and the Wasserstein-1 distance from $0.1122$ to $0.0653$. The mean absolute error decreases from $0.0552$ to $0.0441$, while the relative standard-deviation error decreases from $0.0926$ to $0.0669$. Thus, the jump perturbations improve the overall approximation rather than merely increasing the variance of the generated samples.

The improvement is substantially more pronounced in the tails. On the right tail, the log-CCDF RMSE decreases from $0.4161$ to $0.1556$, while the relative error of the $0.995$ quantile decreases from $0.1697$ to $0.0301$. The absolute error in probability mass beyond the target $0.995$ quantile is reduced from $1.75 \times 10^{-3}$ to $3.0 \times 10^{-4}$, and the Hill-index error decreases from $0.2497$ to $0.0450$. These results indicate that the baseline flow substantially underestimates the extreme right tail, whereas the jump-augmented sampler recovers both its probability mass and its decay rate much more accurately.

Table 5: **Distribution fitting results on the heavy-tailed toy target.** We report overall distributional discrepancies and tail-sensitive errors. For the tail metrics, the right and left tails are evaluated separately. Lower is better for all reported error and distance metrics.

| Metric | Baseline ODE | ODE + Jumps |
|---|---|---|
| *Overall distribution fitting* | | |
| KS distance ↓ | 0.0181 | **0.0129** |
| Wasserstein-1 distance ↓ | 0.1122 | **0.0653** |
| Mean absolute error ↓ | 0.0552 | **0.0441** |
| Std. relative error ↓ | 0.0926 | **0.0669** |
| *Right-tail fitting* | | |
| Log-CCDF RMSE ↓ | 0.4161 | **0.1556** |
| $Q_{0.995}$ relative error ↓ | 0.1697 | **0.0301** |
| Tail-mass abs. error ($q = 0.995$) ↓ | 0.00175 | **0.00030** |
| Hill-index error ↓ | 0.2497 | **0.0450** |
| *Left-tail fitting* | | |
| Log-CCDF RMSE ↓ | 0.1975 | **0.1539** |
| $Q_{0.005}$ relative error ↓ | 0.0734 | **0.0609** |
| Tail-mass abs. error ($q = 0.005$) ↓ | 0.00090 | **0.00072** |
| Hill-index error ↓ | 0.0879 | **0.0410** |

A similar trend holds for the left tail. The log-CCDF RMSE decreases from $0.1975$ to $0.1539$, the relative error of the $0.005$ quantile decreases from $0.0734$ to $0.0609$, and the tail-mass error decreases from $9.0 \times 10^{-4}$ to $7.2 \times 10^{-4}$. The Hill-index error is also reduced from $0.0879$ to $0.0410$. Although the improvement is smaller than on the right tail, JumpSampler consistently yields a closer approximation across all reported tail-sensitive metrics.

Overall, these results support the mechanism illustrated in Figure 2: a Gaussian-driven flow can closely reproduce the bulk of a heavy-tailed target while still exhibiting a systematic deficiency in low-probability regions. Sparse Student's $t$ perturbations specifically alleviate this tail mismatch and improve extreme-event coverage, while simultaneously preserving, and in this experiment improving, the global distributional fit.

### C.2 Results on GenEval Benchmark

Table 6: GenEval results on text-to-image generation. We compare the baseline sampler with JumpSampler on SD3-M and FLUX.1 across different compositional categories.

| Model | Sampler | Single↑ | Two↑ | Counting↑ | Colors↑ | Position↑ | Color-Attr.↑ | **Overall↑** |
|---|---|---|---|---|---|---|---|---|
| SD3-M | Baseline | 99.69 | 87.63 | 69.38 | **84.57** | 28.00 | 63.75 | 72.17 |
| SD3-M | Ours ($\beta = 0.1, \lambda = 1.0$) | 99.69 | 87.63 | **69.69** | 84.31 | **28.50** | **65.25** | **72.51** |
| FLUX.1 | Baseline | 97.50 | 79.29 | 77.19 | 80.05 | 18.50 | 39.25 | 65.30 |
| FLUX.1 | Ours ($\beta = 2.5, \lambda = 1.0$) | 97.50 | **79.55** | **77.50** | 80.05 | 18.50 | **39.50** | **65.43** |

We further evaluate JumpSampler on GenEval (Ghosh et al., 2023) to examine whether the proposed distribution-level modification preserves the semantic capabilities of pretrained text-to-image models. As shown in Table 6, JumpSampler achieves comparable or slightly improved performance over the baseline on both SD3-M and FLUX.1. Most compositional categories are either preserved or moderately improved, and the overall score increases for both models.

We emphasize that JumpSampler is not explicitly designed to optimize semantic alignment. Its primary effect is to modify the sampling distribution through sparse heavy-tailed perturbations, rather than altering the learned conditional representation or introducing an additional semantic guidance objective. Therefore, only modest changes in semantic benchmark scores are expected. From this perspective, the GenEval results mainly serve as a generalization and compatibility check: despite modifying the sampling dynamics at the distributional level, JumpSampler does not induce a

noticeable degradation in semantic fidelity and remains compatible with the original text-conditioning capability of different pretrained models.

### C.3 Robustness across random seeds

Table 7 reports multi-seed evaluation on ImageNet-256. The FID improvement is consistently observed across different random seeds, while IS remains comparable to the baseline with moderate variation. These results indicate that the improvement of JumpSampler is reproducible and is not driven by a single favorable sampling run.

Table 7: **Multi-seed evaluation on ImageNet-256.** We report FID and IS over three random seeds for the baseline sampler and JumpSampler. The last column reports the mean and standard deviation.

| Metric | Method | Seed 0 | Seed 1 | Seed 2 | Mean $\pm$ Std. |
|---|---|---|---|---|---|
| FID $\downarrow$ | Baseline | 1.431 | 1.444 | 1.427 | $1.434 \pm 0.007$ |
| | JumpSampler | **1.390** | **1.430** | **1.416** | $\mathbf{1.412 \pm 0.017}$ |
| IS $\uparrow$ | Baseline | 305.06 | 303.03 | 305.46 | $304.52 \pm 1.06$ |
| | JumpSampler | **305.70** | **306.03** | **306.41** | $\mathbf{306.05 \pm 0.29}$ |

### C.4 Sensitivity to jump intensity and temporal scheduling

We further study the sensitivity of JumpSampler to the jump intensity $\lambda$ and temporal exponent $\beta$ in Tables 8 and 9. For the jump intensity, a moderate value of $\lambda$ generally provides the best trade-off. A smaller jump intensity performs competitively at the smallest evaluation scales, whereas the default $\lambda = 1$ achieves better centered ECS over most of the remaining range. Increasing the jump intensity further gradually degrades the centered ECS, suggesting that overly frequent jumps can perturb the original sampling trajectory and offset the benefit of tail exploration.

The results are also sensitive to the temporal scheduling of the jumps. Among the tested values, the default $\beta = 2.4$ consistently yields the lowest centered ECS across the evaluation scales, while both smaller and larger exponents lead to weaker performance. This indicates that concentrating the jump perturbations according to an appropriate temporal profile is important: insufficient or overly aggressive temporal decay can reduce the benefit of the jump mechanism. Overall, these ablations support the default choice of $\lambda = 1$ and $\beta = 2.4$ used in our main experiments.

Table 8: **Ablation on the jump intensity $\lambda$ on ImageNet-256 with SiT-XL/2.** We vary the jump intensity $\lambda$ while fixing $\beta = 2.4$ and $\nu = 3$. Baseline denotes sampling without jumps. The default setting is $\lambda = 1$. Lower ECS is better.

| | Raw ECS ($\times 10^{-3}$) $\downarrow$ | | | | | Centered ECS ($\times 10^{-3}$) $\downarrow$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $T$ | Baseline | $\lambda = 0.5$ | $\lambda = 1$ | $\lambda = 2$ | $\lambda = 5$ | Baseline | $\lambda = 0.5$ | $\lambda = 1$ | $\lambda = 2$ | $\lambda = 5$ |
| 0.02 | 5.282 | 5.570 | 5.278 | 5.298 | 5.375 | 0.0448 | **0.0434** | 0.0435 | 0.0438 | 0.0453 |
| 0.05 | 5.283 | 5.571 | 5.279 | 5.299 | 5.376 | 0.1120 | **0.1086** | 0.1088 | 0.1096 | 0.1133 |
| 0.10 | 5.286 | 5.572 | 5.282 | 5.301 | 5.379 | 0.2241 | **0.2174** | 0.2177 | 0.2192 | 0.2266 |
| 0.20 | 5.292 | 5.574 | 5.286 | 5.305 | 5.383 | 0.4477 | 0.4353 | **0.4352** | 0.4381 | 0.4528 |
| 0.50 | 5.281 | 5.544 | 5.273 | 5.286 | 5.363 | 1.0940 | 1.0732 | **1.0677** | 1.0748 | 1.1097 |
| 0.70 | 5.239 | 5.486 | 5.230 | 5.239 | 5.313 | 1.4845 | 1.4645 | **1.4531** | 1.4628 | 1.5087 |
| 1.00 | 5.129 | 5.348 | 5.118 | 5.122 | 5.185 | 1.9912 | 1.9753 | **1.9551** | 1.9678 | 2.0249 |
| 2.00 | 4.511 | 4.643 | 4.496 | 4.494 | 4.513 | 3.0449 | 3.0508 | **2.9955** | 3.0116 | 3.0835 |
| 3.00 | 3.774 | 3.859 | 3.776 | 3.771 | 3.770 | 3.4083 | 3.4485 | **3.3604** | 3.3746 | 3.4411 |

### C.5 Generalization to Pixel-Space Second-Order Sampling

To further evaluate the generality of JumpSampler, we extend it beyond the latent-space and first-order sampling settings considered in the main experiments. Specifically, we apply JumpSampler to the

Table 9: **Ablation on the temporal exponent $\beta$ on ImageNet-256 with SiT-XL/2.** We vary $\beta$, which controls the temporal profile of the jump process, while fixing $\lambda = 1$ and $\nu = 3$. Baseline denotes sampling without jumps. The default setting is $\beta = 2.4$. Lower ECS is better.

| | Raw ECS ($\times 10^{-3}$) ↓ | | | | Centered ECS ($\times 10^{-3}$) ↓ | | | |
|---|---|---|---|---|---|---|---|---|
| $T$ | Baseline | $\beta = 2.0$ | $\beta = 2.4$ | $\beta = 3.0$ | Baseline | $\beta = 2.0$ | $\beta = 2.4$ | $\beta = 3.0$ |
| 0.02 | 5.282 | 5.386 | 5.278 | 5.436 | 0.0448 | 0.0444 | **0.0435** | 0.0449 |
| 0.05 | 5.283 | 5.388 | 5.279 | 5.437 | 0.1120 | 0.1110 | **0.1088** | 0.1122 |
| 0.10 | 5.286 | 5.390 | 5.282 | 5.439 | 0.2241 | 0.2220 | **0.2177** | 0.2245 |
| 0.20 | 5.292 | 5.393 | 5.286 | 5.443 | 0.4477 | 0.4435 | **0.4352** | 0.4486 |
| 0.50 | 5.281 | 5.372 | 5.273 | 5.426 | 1.0940 | 1.0853 | **1.0677** | 1.0993 |
| 0.70 | 5.239 | 5.321 | 5.230 | 5.379 | 1.4845 | 1.4751 | **1.4531** | 1.4952 |
| 1.00 | 5.129 | 5.193 | 5.118 | 5.258 | 1.9912 | 1.9818 | **1.9551** | 2.0101 |
| 2.00 | 4.511 | 4.530 | 4.496 | 4.601 | 3.0449 | 3.0362 | **2.9955** | 3.0819 |
| 3.00 | 3.783 | 3.787 | 3.776 | 3.846 | 3.4083 | 3.4107 | **3.3604** | 3.4557 |

pretrained **EDM2-S** (Karras et al., 2024) model on **ImageNet-64**, where generation is performed directly in pixel space using the original second-order sampler. This provides a complementary setting for examining whether the proposed jump mechanism remains compatible with a different model family, state space, and higher-order numerical solver.

Together with our latent-space experiments, this result suggests that JumpSampler is not tied to a particular latent representation or first-order Euler discretization, and can be incorporated into substantially different sampling pipelines as an inference-time modification.

Table 10: **Generalization to second-order pixel-space sampling on ImageNet-64.** We evaluate JumpSampler on the pretrained **EDM2-S** model using its second-order pixel-space sampler. We report FID and FD-DINOv2 (Ahn, 2024) against the official ImageNet-64 reference statistics. Despite the already strong baseline, JumpSampler preserves generation quality and yields slightly lower values on both metrics.

| Method | FID ↓ | FD-DINOv2 ↓ |
|---|---|---|
| EDM2-S Baseline | 1.04354 | 56.3716 |
| EDM2-S + JumpSampler ($\beta = 3.0, \lambda = 1.0, \nu = 4$) | **1.04337** | **56.3622** |

### C.6 Additional Visual Results

Figure 3 shows additional visual results using our method on T2I and SiT models.

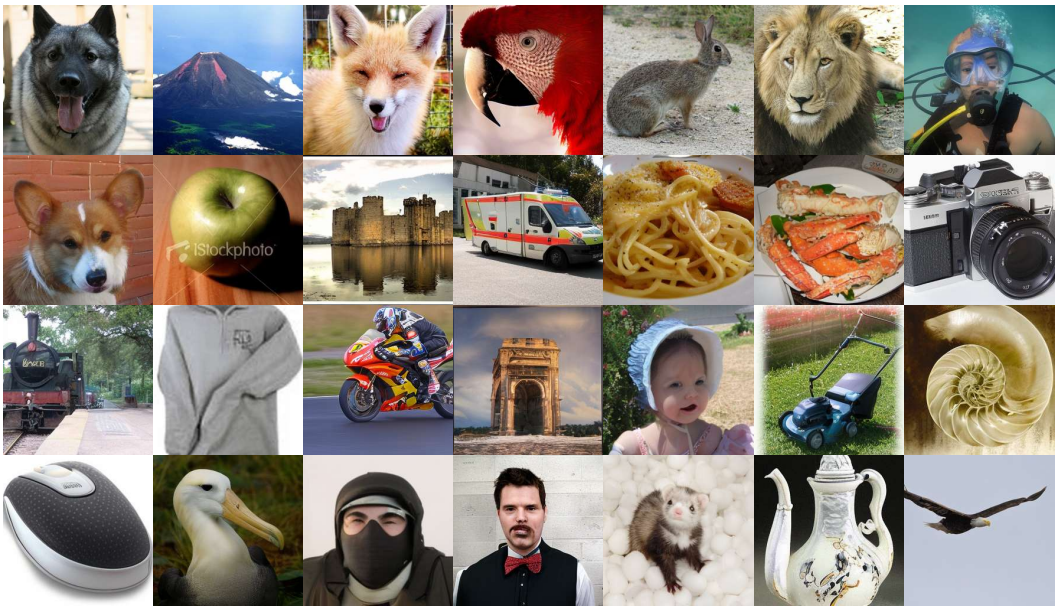


Images generated by SD3 with our method.



Images generated by FLUX with our method.

(a) Images generated by the FLUX and SD3.



(b) Images generated by the SiT-XL/2 (REPA) model with our method.

Figure 3: Additional visual results on different models.