DENOISING LÉVY SCORES FOR GENERATIVE MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

We investigate Lévy-driven sampling to improve sample quality in score-based generative modeling under a fixed function-evaluation budget. Lévy jumps enable long-range transitions between separated regions, offering an alternative to Brownian Langevin relaxation at each noise level. Unlike existing Lévy-based generative models, which introduce jumps through non-Gaussian forward noising, we retain Gaussian perturbations and introduce *Lévy Langevin correctors* (LLCs) only at sampling step. These correctors depend only on the Lévy score, a nonlocal analogue of the Stein score. We develop *denoising Lévy score matching* (DLSM) by perturbing the data with different levels of Gaussian noise and jointly estimating the corresponding Lévy scores. On synthetic multimodal and heavy-tailed target distributions, our method achieves up to a $9.5\times$ reduction in W_1 error relative to Brownian samplers at matched NFEs. At matched low-NFE budgets, our method achieves lower FID than Brownian samplers on CIFAR-10, CelebA-64, and CIFAR-10-LT; on CIFAR-10-LT, it also reproduces the long-tailed class distribution more accurately.

1 INTRODUCTION

Score-based diffusion models have become a leading approach to generative modeling, with strong performance across images, audio, and other high-dimensional domains (Song & Ermon, 2019; Ho et al., 2020; Song et al., 2021b). They gradually perturb the data distribution toward a simple reference distribution, estimate the Stein scores of the resulting noisy marginals, and generate samples by reversing this process. Gaussian noising makes the main ingredients particularly simple: the transition density is available in closed form, denoising score matching reduces score estimation to regression with an explicit conditional target, and sufficiently strong noising provides a simple Gaussian initialization for generation (Vincent, 2011; Song & Ermon, 2019; Song et al., 2021b).

Despite these advantages, high-quality generation often requires many sequential evaluations of the score network (Song et al., 2021a; Lu et al., 2022; Karras et al., 2022). Much of the fast-sampling literature keeps the forward probability path fixed and changes the dynamics used for generation. For a fixed diffusion path, the reverse-time SDE, probability-flow ODE, and more general Markov processes can realize the same marginal distributions (Song et al., 2021b; Holderrieth et al., 2025). This freedom can be exploited at finite sampling budgets: DDIM varies the stochasticity used for generation, while higher-order solvers and improved discretizations reduce the number of function evaluations required by the chosen dynamics (Song et al., 2021a; Lu et al., 2022; Karras et al., 2022). Even when the continuous-time marginals agree, score error and discretization can lead to substantially different finite-step behavior (Cao et al., 2023).

Most of fast finite-step solvers do not, however, change the local structure of the standard Gaussian diffusion dynamics. The learned Stein score $S[p_t](x) = \nabla_x \log p_t(x)$ is a local differential quantity, and the commonly used Brownian and deterministic trajectories are continuous; neither provides a direct long-range move between separated regions of the state space. Conventional diffusion models have also been observed to lose fidelity and diversity on imbalanced data, including mode-collapse behavior (Deasy et al., 2021; Yoon et al., 2023; Shariatian et al., 2025). Lévy-driven Langevin samplers instead use discontinuous, heavy-tailed jumps to move across low-density regions, and have demonstrated improved exploration of multimodal targets (Şimşekli, 2017; Huang et al., 2021; Oechsler, 2024). Motivated by this nonlocal exploration, another line of generative work

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.

changes the forward perturbation law itself. Gaussian perturbations have been replaced by non-Gaussian noise, such as Gamma, generalized-Gaussian, Student-t, α-stable, and compound-Poisson noise (Nachmani et al., 2021; Deasy et al., 2021; Pandey et al., 2024; Yoon et al., 2023; Shariatian et al., 2025; Baule, 2026). Once a perturbation law is chosen, its forward transition, denoising target or generalized "score", reference distribution, and reverse dynamics must be derived for that choice. These non-Gaussian models can substantially improve sample diversity, tail coverage, and robustness on imbalanced data, and potentially can improve finite-step sampling (Yoon et al., 2023; Shariatian et al., 2025; Popov et al., 2025).

Existing non-Gaussian diffusion models typically require committing to a special forward noise family so that key ingredients—transition kernels, denoising targets, and the terminal reference distribution—remain analytically tractable. As a result, the learned score, the training objective, and the reverse-time generative dynamics are tightly coupled to that chosen family. This dependence creates a practical bottleneck: introducing a non-Gaussian perturbation different from the standard Gaussian forward process is possible only for a limited set of noise distributions, and often forces a full redesign of the modeling pipeline. In particular, changing the forward noise generally requires redefining the terminal reference distribution, deriving new denoising or score-matching targets, and re-deriving the reverse-time dynamics. These steps can substantially increase complexity, for example through more involved transition-kernel evaluations, less convenient reference sampling, and bespoke learning targets.

These limitations motivate a fundamental question: *Can we retain the standard Gaussian forward process while constructing reverse-time generative dynamics that incorporate Lévy-type noise with an arbitrary prescribed Lévy measure ν, subject to suitable admissibility conditions, and preserve the prescribed time-reversed marginals?*

We address this question by proposing the *Gaussian–Lévy Score Model* (GLSM), which retains the standard Gaussian forward diffusion while incorporating a prescribed Lévy measure into the generative dynamics through a Lévy-Ito SDE with Lévy Langevin correctors (LLCs). The Gaussian forward process provides the same tractable ingredients used by standard score-based models: an explicit transition density, a denoising regression objective, and a simple reference distribution for initialization. Given any Lévy measure ν, GLSM constructs Lévy jump-diffusion reverse dynamics whose time marginals match those of the Gaussian forward diffusion. The resulting dynamics are driven by the stein diffusion score and our new Lévy score $S_\nu[p_t]$. Leveraging the Gaussian transition density, we derive conditional targets for learning the Lévy score $S_\nu[p_t]$ from the same clean–noisy pairs used in standard denoising score matching, and show that the associated conditional mean equals $S_\nu[p_t]$. This leads to *denoising Lévy score matching* (DLSM). Finally, the learned Lévy score is used to build Lévy Langevin correctors (LLCs) that target the Gaussian-smoothed marginals and are annealed over decreasing diffusion times during generation.

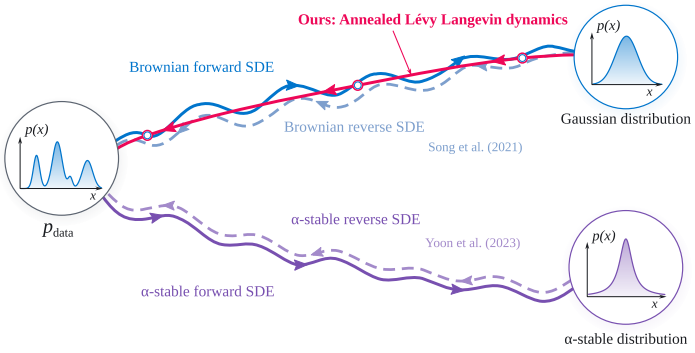


Figure 1: **Forward noising and generative dynamics.** Standard score-based diffusion uses Gaussian noising with Brownian reverse dynamics, whereas LIM (Yoon et al., 2023) uses α-stable forward noising with Lévy-driven reverse dynamics. We keep the Gaussian forward process and use Lévy generative dynamics whose jump measure can be prescribed independently.

Contributions. Our contributions are summarized as follows:

- **Gaussian–Lévy Score Model.** We introduce a score-based generative model that retains the standard Gaussian forward diffusion while allowing reverse-time generative dynamics governed by a Lévy–Itô SDE with both diffusion and jump components (Theorem 3.1).

Central to our construction is the Lévy score $S_\nu[p_t]$, a nonlocal analogy to the diffusion score, which provides the drift correction needed to match – *for any prescribed Lévy measure* ν – the time-marginal distributions of the standard Gaussian forward diffusion. When $\nu = 0$, our framework recovers the classical reverse-time SDE and probability-flow ODE; when $\nu \neq 0$, it produces discontinuous (jump) trajectories while preserving the same Gaussian marginal path.

- **Denoising Lévy score matching.** We develop DLSM, a denoising score-matching objective for learning the Lévy score $S_\nu[p_t]$ from Gaussian perturbations (Theorem 3.2). DLSM takes the form of a (Bregman) regression objective and can be trained using exactly the same clean–noisy pairs as in standard denoising score matching, with $X_0 \sim p_{\text{data}}$ and $X_t \sim p_{0t}(\cdot \mid X_0)$, without requiring non-Gaussian forward noising or alternative reference distributions.
- **Annealed Lévy corrector.** We construct *pure-jump* Lévy SDE whose invariant distribution is the Gaussian-smoothed marginal p_t, using the learned Lévy score (Theorem 3.3). By annealing these correctors over a decreasing sequence of diffusion times, we obtain the full GLSM sampling algorithm (Algorithm 1).
- **Finite-budget generation.** Across controlled multimodal and heavy-tailed targets and image-generation benchmarks, GLSM achieves improved mode recovery and sample quality at matched function-evaluation budgets. On imbalanced image datasets, GLSM further improves support coverage and class-conditional fidelity.

2 Gaussian and Lévy Diffusion Models

2.1 Score-Based Diffusion Models

GLSM retains the Gaussian forward process together with the learning problem induced by its transition density. A continuous-time score-based model perturbs data with the forward SDE

$$\mathrm{d}X_t = f(X_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}W_t, \qquad X_0 \sim p_0 = p_{\text{data}}. \tag{1}$$

For the variance-exploding (VE), variance-preserving (VP), and related linear SDEs used in practice, the transition density is Gaussian,

$$p_{0t}(x_t \mid x_0) = \mathcal{N}(x_t; \alpha_t x_0, \sigma_t^2 I), \qquad X_t = \alpha_t X_0 + \sigma_t \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I). \tag{2}$$

The coefficients are chosen so that the terminal marginal p_T is close to a known Gaussian reference distribution. Although the marginal density $p_t(x) = \int p_0(x_0) p_{0t}(x \mid x_0)\,\mathrm{d}x_0$ is generally unavailable in closed form, (2) allows direct sampling of X_t from any data point X_0.

For Gaussian diffusion, the reverse-time SDE depends on the unknown marginal only through its Stein score. For a positive density p, let

$$S[p](x) := \nabla_x \log p(x) \tag{3}$$

denote the Stein score, which we refer to below simply as the score. Under the standard regularity assumptions, the reverse-time SDE is (Anderson, 1982; Song et al., 2021b)

$$\mathrm{d}X_t = \big[f(X_t, t) - g(t)^2 S[p_t](X_t)\big]\,\mathrm{d}t + g(t)\,\mathrm{d}\widetilde{W}_t, \qquad t : T \to 0, \tag{4}$$

where $\mathrm{d}t < 0$. Estimating $S[p_t]$ is therefore sufficient to simulate the reverse process in (4).

Gaussian noising makes this estimation problem a regression problem. The conditional score is explicit, $\nabla_{x_t} \log p_{0t}(x_t \mid x_0) = -(x_t - \alpha_t x_0)/\sigma_t^2$, and satisfies

$$\mathbb{E}[\nabla_{x_t} \log p_{0t}(X_t \mid X_0) \mid X_t = x] = S[p_t](x). \tag{5}$$

Denoising score matching (DSM) therefore trains a time-conditioned network by minimizing

$$\mathcal{L}_{\text{DSM}}(\theta) = \mathbb{E}\left[\lambda(t)\,\|S_\theta(X_t, t) - \nabla_{x_t} \log p_{0t}(X_t \mid X_0)\|^2\right], \tag{6}$$

where t is drawn from the training distribution, $X_0 \sim p_{\text{data}}$, and $X_t \sim p_{0t}(\cdot \mid X_0)$ (Vincent, 2011; Song & Ermon, 2019; Song et al., 2021b). The minimizer of (6) satisfies $S_{\theta^\star}(x, t) = S[p_t](x)$ almost surely. GLSM keeps this Gaussian forward process, including its transition densities, noisy marginals p_t, and Gaussian terminal distribution. The same Gaussian marginals also admit different generative dynamics: the probability-flow ODE shares the marginals of the reverse SDE, while at a fixed diffusion time the score drives Langevin dynamics with invariant distribution p_t.

2.2 Lévy–Itô Diffusion Models

Lévy–Itô diffusion models introduce nonlocal dynamics by changing the forward perturbation process itself. The Lévy–Itô Model (LIM) replaces Brownian motion by an isotropic α-stable Lévy process L_t^α (Yoon et al., 2023),

$$\mathrm{d}X_t = \mu(X_{t-}, t)\,\mathrm{d}t + c(t)\,\mathrm{d}L_t^\alpha, \qquad 1 < \alpha < 2.$$

The conditional transition law is then α-stable rather than Gaussian, and sufficiently strong noising leads to an α-stable reference distribution. Thus the forward perturbation law changes both the conditional kernel used for learning and the distribution used to initialize generation. The score needed by LIM changes with the forward process as well. Its reverse dynamics depends on the fractional score

$$S_t^{(\alpha)}(x) := \frac{(-\Delta)^{(\alpha-2)/2}\nabla p_t(x)}{p_t(x)}, \tag{7}$$

which is learned from α-stable perturbations by fractional denoising score matching (Yoon et al., 2023).

The exact time-reversal representation derived for LIM (Yoon et al., 2023) is

$$\mathrm{d}\bar{X}_t = \left[\mu(\bar{X}_{t+}, t) - \alpha c(t)^\alpha S_t^{(\alpha)}(\bar{X}_{t+})\right]\mathrm{d}t + c(t)\,\mathrm{d}\bar{L}_t^\alpha + \mathrm{d}\bar{Z}_t, \qquad t : T \to 0, \tag{8}$$

In (8), $\bar{L}_t^\alpha$ is an α-stable Lévy process in reverse time and $\bar{Z}_t$ is a data-dependent finite-variation process whose law depends on density ratios. The practical stochastic sampler in LIM omits $\mathrm{d}\bar{Z}_t$, yielding an approximate reverse SDE that can be simulated with the learned fractional score (Yoon et al., 2023). Thus in LIM, the α-stable transition law, fractional score, reference distribution, and reverse process all arise from the same α-stable forward noising process. GLSM keeps the Gaussian forward process instead and introduces the Lévy measure only through generation.

3 Gaussian–Lévy Score Model

In our method, we use the same Gaussian forward SDE (1) as in the conventional score-based diffusion models, and denote its marginals as $p_t,\ 0 \le t \le T$. The standard diffusion score mathching provides the Stein score $S[p_t]$ required by the classic Gaussian reverse dynamics. In our new method, however, we use an arbitrary Lévy jump process (e.g., compound Poisson process and $alpha$-stable process) in the reversed dynamics by introducing the new nonlocal score $S_{\nu_t}[p_t]$. In this sectoin, we first characterize the reverse jump-diffusions compatible with the Gaussian marginal path by introducing the Lévy Langevin corrector, then show how to estimate the Lévy score from Gaussian perturbation data. For sampling, we use the learned Lévy score in fixed-target pure-jump Lévy Langevin dynamics and anneal these dynamics across the Gaussian marginals.

3.1 Lévy Jump-Diffusion Reverse Dynamics

We begin with the jump noise to be used during generation. Let ν be a state-independent Lévy measure satisfying $\int_{\mathbb{R}^d\setminus\{0\}}(1 \wedge \|r\|^2)\,\nu(\mathrm{d}r) < \infty$. For a smooth test function φ, its jump generator is

$$\mathcal{L}_\nu\varphi(x) = \int_{\mathbb{R}^d\setminus\{0\}} \left[\varphi(x+r) - \varphi(x) - \chi(r)\, r^\top \nabla\varphi(x)\right]\nu(\mathrm{d}r),$$

where χ is the truncation indicator function in the Lévy–Khintchine representation. For finite Lévy measures we use the equivalent uncompensated convention $\chi \equiv 0$ after the usual drift adjustment (Applebaum, 2009).

To construct a reverse process with a prescribed jump measure, we need to express its contribution to the density evolution in terms of p. With a given Lévy measure, ν, Huang et al. (2026; 2025) define the *Lévy score* for each positive smooth density p

$$S_\nu[p](x) := -\frac{1}{p(x)}\int_{\mathbb{R}^d\setminus\{0\}} r\left[\int_0^1 p(x-\eta r)\,\mathrm{d}\eta - \chi(r)p(x)\right]\nu(\mathrm{d}r). \tag{9}$$

For isotropic α-stable noise, (9) reduces, up to the normalization of the stable generator, to (7). We apply the same definition pointwise in time to the prescribed family $\{\nu_t\}_{0\le t\le T}$.

Let $h(s)$ denote the Brownian diffusion amplitude used at generation time, and write the time-reversed Lévy measure by $\overleftarrow{\nu}_s := \nu_{T-s}$. We consider the following generative dynamics on $0 \le s \le T$:

$$\mathrm{d}X_s = b(X_{s-}, s)\,\mathrm{d}s + h(s)\,\mathrm{d}\widetilde{W}_s + \mathrm{d}J_s. \tag{10}$$

where J_s is a pure-jump process whose compensator is $\overleftarrow{\nu}_s(\mathrm{d}r)\,\mathrm{d}s$.

Let ρ_s denote the density of the Lévy-Ito SDE (10). Then ρ satisfies the Lévy-Fokker-Planck equation

$$\partial_s \rho_s = -\nabla \cdot \left[\rho_s \left(b - \frac{h(s)^2}{2} S[\rho_s] - S_{\overleftarrow{\nu}_s}[\rho_s]\right)\right]. \tag{11}$$

Meanwhile, the time-reversed density function $\overleftarrow{p}_s$ for the forward Gaussian SDE (1) satisfies

$$\partial_s \overleftarrow{p}_s = -\nabla \cdot \left[\overleftarrow{p}_s \left(-f(\cdot, T-s) + \frac{g(T-s)^2}{2} S[\overleftarrow{p}_s]\right)\right]. \tag{12}$$

Matching (11) with (12) determines the drift b as the *Lévy Langevian corrector* . This yields a family of reverse processes indexed directly by the generation-time noise $h(s)$ and $\overleftarrow{\nu}_s$.

Theorem 3.1 (Lévy jump-diffusion reverse dynamics)**.** *Let $(p_t)_{0\le t\le T}$ be the density functions for the solution to the Itô SDE (1). Assume for each $s \in (0,T)$, $\overleftarrow{p}_s := p_{T-s}$ is positive and sufficiently regular for $S[\overleftarrow{p}_s]$ and $S_{\overleftarrow{\nu}_s}[\overleftarrow{p}_s]$ to be well defined, and that the forward equation associated with the SDE below is well posed and unique in a class containing $\{\overleftarrow{p}_s : 0 \le s \le T\}$. For any measurable Brownian amplitude $h(s)$ and any admissible family of Lévy measures $\overleftarrow{\nu}_s$ satisfying the integrability conditions in Appendix A, define the following Lévy-Ito SDE with the initial $Y_0 \sim p_T$:*

$$\mathrm{d}Y_s = \left[-f(Y_{s-}, T-s) + \frac{g(T-s)^2 + h(s)^2}{2} S[\overleftarrow{p}_s](Y_{s-}) + S_{\overleftarrow{\nu}_s}[\overleftarrow{p}_s](Y_{s-})\right]\mathrm{d}s + h(s)\,\mathrm{d}\widetilde{W}_s + \mathrm{d}J_s, \tag{13}$$

where the jump process J has compensator $\overleftarrow{\nu}_s(\mathrm{d}r)\,\mathrm{d}s$. Then $\mathrm{Law}(Y_s) = \overleftarrow{p}_s = p_{T-s}$, $0 \le s \le T$.

The Lévy measure $\overleftarrow{\nu}_s$ and Brownian amplitude $h(s)$ above can be specified by the user as the freedom of the design strategy. In particular, $\overleftarrow{\nu}_s = 0$ and $h(s) = g(T-s)$ recovers the standard reverse-time SDE, and $\overleftarrow{\nu}_s = 0$ and $h(s) = 0$ gives the probability-flow ODE. A nonzero $\overleftarrow{\nu}_s$ gives discontinuous trajectories without changing the Gaussian marginals. As in existing Gaussian ODE/SDE samplers and same-marginal reverse dynamics in Lévy models, different generation dynamics can have different finite-step behavior after discretization (Cao et al., 2023; Popov et al., 2025).

3.2 Denoising Lévy Score Matching

To implement (13), we need to estimate the reverse-time Lévy score $S_{\overleftarrow{\nu}_s}[\overleftarrow{p}_s]$. Since $\overleftarrow{p}_s = p_{T-s}$ and $\overleftarrow{\nu}_s = \nu_{T-s}$, it is more convenient to consider the forward-time score $S_{\nu_t}[p_t]$ with $t = T-s$. We estimate $S_{\nu_t}[p_t]$ from Gaussian perturbations, where $p_{0t}(\cdot \mid x_0)$ is known. Consider the *conditional Lévy score* $T_{\nu_t}(x_t, x_0; t) := S_{\nu_t}[p_{0t}(\cdot \mid x_0)](x_t)$. The key fact is that, for fixed ν_t, $p \mapsto p\,S_{\nu_t}[p]$ is a linear map. Applying this linearity to the Gaussian mixture $p_t(x) = \int p_0(x_0) p_{0t}(x \mid x_0)\,\mathrm{d}x_0$ gives

$$p_t(x) S_{\nu_t}[p_t](x) = \int p_0(x_0) p_{0t}(x \mid x_0) T_{\nu_t}(x, x_0; t)\,\mathrm{d}x_0. \tag{14}$$

After division by $p_t(x)$, (14) becomes a conditional-expectation identity directly analogous to equality (5) in DSM.

Theorem 3.2 (Denoising Lévy-score identity)**.** *Under the integrability conditions in Appendix B,*

$$\mathbb{E}[T_{\nu_t}(X_t, X_0; t) \mid X_t = x] = S_{\nu_t}[p_t](x) \tag{15}$$

for p_t-almost every x.

The identity turns Lévy-score estimation into the same kind of conditional regression problem used by DSM. Let D be a Bregman divergence on $\mathbb{R}^d$ and train S_θ^ν by minimizing

$$\mathcal{L}_{\mathrm{DLSM}}(\theta) = \mathbb{E}\left[w(t)D(T_{\nu_t}(X_t, X_0; t), S_\theta^\nu(X_t, t))\right], \tag{16}$$

where the expectation uses the same training draws as in DSM (6): diffusion time t, $X_0 \sim p_{\mathrm{data}}$, and $X_t \sim p_{0t}(\cdot \mid X_0)$. Bregman regression is minimized by the conditional mean (Banerjee et al., 2005), so Theorem 3.2 gives the minimizer $S_{\theta^\star}^\nu(x,t) = S_{\nu_t}[p_t](x)$ almost surely whenever (16) is finite. Choosing a different Lévy measure changes the conditional target T_{ν_t} but not the Gaussian sampling procedure; the divergence D can be selected according to the moment properties of the target.

Compound-Poisson jumps. For a compound-Poisson measure $\nu_t(\mathrm{d}r) = \lambda_t \rho_t(\mathrm{d}r)$, we adopt the uncompensated jump convention, i.e., $\chi \equiv 0$. Writing $\varepsilon = (X_t - \alpha_t X_0)/\sigma_t \sim \mathcal{N}(0, I)$, the Gaussian transition density gives the conditional Lévy score

$$T_{\nu_t}(\varepsilon, t) = -\lambda_t \int_{\mathbb{R}^d} r \int_0^1 \exp\left(\frac{\eta r^\top \varepsilon}{\sigma_t} - \frac{\eta^2 \|r\|^2}{2\sigma_t^2}\right) \mathrm{d}\eta \, \rho_t(\mathrm{d}r). \tag{17}$$

For the isotropic Gaussian jump amplitudes $\rho_t = \mathcal{N}(0, \tau_t^2 I)$. Let $\kappa_t := \tau_t^2/\sigma_t^2$ and $z := \|\varepsilon\|^2/2$. Evaluating both the r- and η-integrals in (17) gives

$$T_{\nu_t}(\varepsilon, t) = -\frac{\lambda_t \sigma_t}{2} e^z z^{-d/2} \left[\gamma\left(\frac{d}{2}, z\right) - \gamma\left(\frac{d}{2}, \frac{z}{1+\kappa_t}\right)\right] \varepsilon, \tag{18}$$

where $\gamma(a, z) = \int_0^z \xi^{a-1} e^{-\xi} \, \mathrm{d}\xi$ is the lower incomplete gamma function, with (18) interpreted by continuity at $\varepsilon = 0$. Thus the conditional target is available in closed form for isotropic Gaussian jump amplitudes. When the quadratic risk is finite, we use the squared Euclidean divergence $D(a, b) = \|a - b\|^2$; Appendix B gives the derivation of (18).

Isotropic α-stable jumps. Theorem 3.2 also applies to infinite-activity measures. For isotropic α-stable jumps with $1 < \alpha < 2$, the Gaussian conditional target has a finite first moment but an infinite second moment,

$$\mathbb{E}\|T_\alpha(X_t, X_0; t)\| < \infty, \qquad \mathbb{E}\|T_\alpha(X_t, X_0; t)\|^2 = \infty.$$

The conditional mean in (15) is therefore well defined, whereas the squared Euclidean loss is not integrable. We instead take $D = D_{\Psi_\delta}$ with $\Psi_\delta(v) = \delta^2(\sqrt{1 + \|v\|^2/\delta^2} - 1)$. Its linear growth makes the expected loss finite under a first-moment condition, while the minimizer remains the conditional mean. Appendix B gives the explicit conditional target, its tail asymptotics, and the integrability proof.

3.3 Annealed Lévy Langevin Dynamics

For practical sampling, we instantiate GLSM using fixed-target pure-jump Lévy Langevin dynamics. For each diffusion time t, we freeze the target marginal p_t and and Lévy measure ν_t and evolve an internal-time Markov process whose drift is given by the Lévy score:

$$\mathrm{d}Z_\tau = S_{\nu_t}[p_t](Z_{\tau-})\,\mathrm{d}\tau + \mathrm{d}L_\tau^{\nu_t}, \quad \tau > 0 \tag{19}$$

where τ denotes the internal simulation time and $L_\tau^{\nu_t}$ is a pure-jump Lévy process with the (time-homogeneous) Lévy measure ν_t. We refer to (19) as the *Lévy Langevin corrector* (LLC), since its dynamics are constructed so that a prescribed p_t. is an invariant (target) distribution. The infinitesimal generator of Z is $(\mathcal{A}_{\nu_t, p_t}\varphi)(x) = S_{\nu_t}[p_t](x)^\top \nabla\varphi(x) + \mathcal{L}_{\nu_t}\varphi(x)$ where $\mathcal{L}_{\nu_t}$ is the generator of the jump process L^{ν_t}.

Theorem 3.3 (Ergodicity of the Lévy Langevin corrector)**.** *Under the regularity conditions in Appendix A, assume (19) is well posed, ψ-irreducible Harris recurrent, and admits an aperiodic skeleton chain. Then p_t is its unique invariant distribution. Moreover, if $(P_\tau)_{\tau \geq 0}$ denotes the associated Markov semi-group for (19), then*

$$\|P_\tau(x, \cdot) - p_t\|_{\mathrm{TV}} \longrightarrow 0, \qquad \tau \to \infty,$$

for every initial state x in the Harris class.

For this fixed-target dynamics, invariance follows directly from the Lévy-score identity:the adjoint generator annihilates p_t,

$$\mathcal{A}^*_{\nu_t,p_t} p_t = -\nabla \cdot (p_t S_{\nu_t}[p_t]) + \mathcal{L}^*_{\nu_t} p_t = 0. \tag{20}$$

Under the regularity and Harris assumptions of Theorem 3.3, (20) identifies p_t as the invariant distribution and yields uniqueness and convergence. This fixed-target Lévy Langevin construction is closely related to Lévy-driven MCMC(Şimşekli, 2017; Oechsler, 2024).

To approach the data distribution, we anneal the fixed-target correctors over a decreasing schedule of diffusion times $T = t_1 > t_2 > \cdots > t_L = t_{\min} > 0$. We initialize with $X \sim \pi \approx p_T$. At each stage t_i, we run the LLC targeting p_{t_i}, initialized from the output at the previous time level, mirroring the warm-start strategy in annealed Langevin sampling (Song & Ermon, 2019; 2020). Since adjacent Gaussian-smoothed marginals correspond to progressively less perturbed targets, the sample produced at t_i provides an effective initialization for the LLC targeting $p_{t_{i+1}}$. The final stage targets $p_{t_{\min}}$, where $t_{\min}$ chosen small enough so that $p_{t_{\min}}$ closely approximates the data distribution. With step size η_i and a Lévy increment $\Delta L_{\eta_i}^{\nu_{t_i}}$, a single Euler update takes the form $X \leftarrow X + \eta_i S^{\nu}_{\theta}(X, t_i) + \Delta L_{\eta_i}^{\nu_{t_i}}$. Algorithm 1 summarizes the resulting sampler.

Algorithm 1 GLSM sampling with annealed Lévy Langevin dynamics

Require: Lévy-score network S^{ν}_{θ}, prior π, diffusion times $\{t_i\}_{i=1}^{L}$, step sizes $\{\eta_i\}_{i=1}^{L}$, numbers of updates $\{M_i\}_{i=1}^{L}$, Lévy measures $\{\nu_{t_i}\}_{i=1}^{L}$

$X \sim \pi$
for $i = 1, \ldots, L$ **do**
  for $m = 1, \ldots, M_i$ **do**
    sample $\Delta L \sim L_{\eta_i}^{\nu_{t_i}}$
    $X \leftarrow X + \eta_i S^{\nu}_{\theta}(X, t_i) + \Delta L$
  end for
end for
return X

4 Numerical Experiments

We evaluate GLSM on controlled distributions and image-generation benchmarks to study three consequences of generation-time Lévy dynamics: recovery of non-Gaussian structures, sampling efficiency at low NFE, and support coverage under class imbalance. For image experiments, GLSM uses the NCSN++ architecture of Song et al. (2021b) and learns the prescribed Lévy score with DLSM from Gaussian perturbations. Each GLSM update requires one network evaluation; implementation details are provided in Appendix D.

4.1 Controlled Recovery of Non-Gaussian Structures

We first consider two controlled targets that isolate the settings in which nonlocal sampling is most visible. A truncated product Cauchy distribution tests recovery of heavy tails, while an imbalanced 3×3 Gaussian mixture adapted from Shariatian et al. (2025) tests recovery of low-mass, separated modes. We compare Brownian SMLD (Song & Ermon, 2019) with GLSM using compound-Poisson and isotropic α-stable jumps, denoted GLSM-CP and GLSM-α.

Target (NFE)	Metric	SMLD	GLSM-CP (ours)	GLSM-α (ours)
Cauchy (32)	sliced $W_1 \downarrow$	1.1130	**0.1180**	0.4000
	tail MSLE $\downarrow$	0.0718	**0.0019**	0.0058
GMM (128)	sliced $W_1 \downarrow$	0.0287	0.0163	**0.0062**
	mode-TV $\downarrow$	0.0414	0.0244	**0.0102**

Table 1: Recovery of heavy tails and imbalanced modes at fixed sampling budgets.

Both prescribed jump laws improve finite-budget recovery, with different choices favored by the two targets. On the Cauchy distribution, GLSM-CP reduces tail MSLE by $38.2\times$ at 32 NFEs; on the imbalanced mixture, GLSM-α reduces mode-TV by $4.1\times$ at 128 NFEs. The corresponding

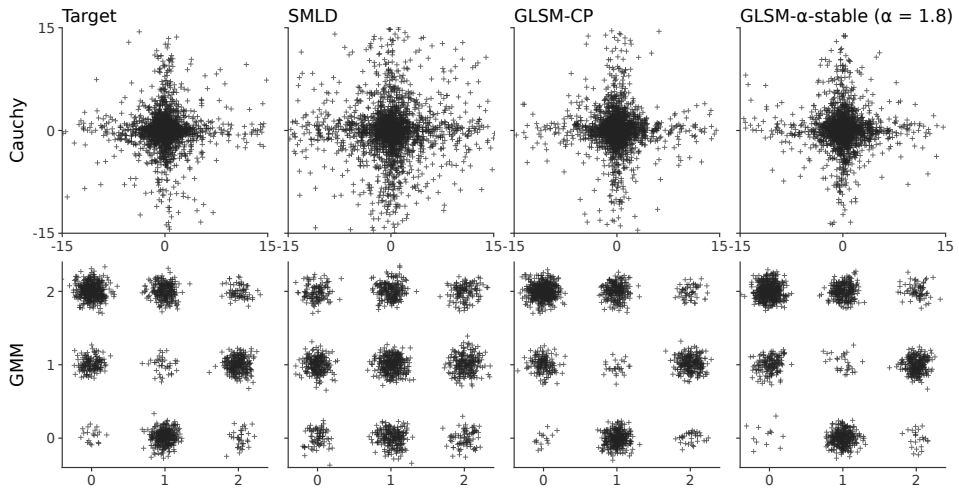


Figure 2: Samples from the controlled targets at the budgets in Table 1. GLSM reproduces the heavy tail of the Cauchy target and the low-mass modes of the imbalanced mixture more accurately.

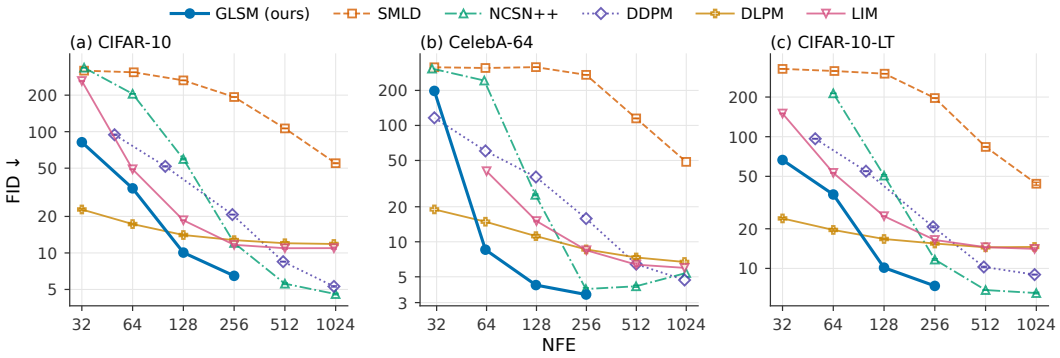


Figure 4: FID versus NFE on image datasets. GLSM reaches lower FID in the low-budget regime on CIFAR-10 and CelebA-64 while using Gaussian perturbations for learning.

sliced W_1 errors decrease by $9.4\times$ and $4.6\times$, showing that the improvement is also visible in the full distribution rather than only in the target-specific diagnostic. The contrast is summarized in Table 1 and Figure 2.

4.2 Sampling Efficiency on Image Data

We next compare GLSM with NCSN++ using predictor–corrector sampling (Song et al., 2021b), DDPM (Ho et al., 2020), LIM (Yoon et al., 2023), and DLPM (Shariatian et al., 2025) on CIFAR-10 and CelebA-64. Figure 4 shows that GLSM improves low-NFE image generation on both datasets, reaching high sample quality with fewer network evaluations. On CIFAR-10, GLSM reaches FID 6.47 at 256 NFEs, while on CelebA-64 it reaches FID 4.24 at 128 NFEs. The FID-versus-NFE curves in Figure 4 show that this improvement persists across the low-budget regime.

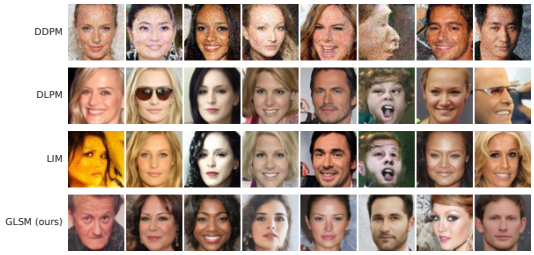


Figure 3: CelebA-64 samples at 64 NFEs.

The same behavior is visible qualitatively on CelebA-64. At 64 NFEs, GLSM already produces coherent face samples, while the competing samplers remain visibly degraded at the same budget (Figure 3).

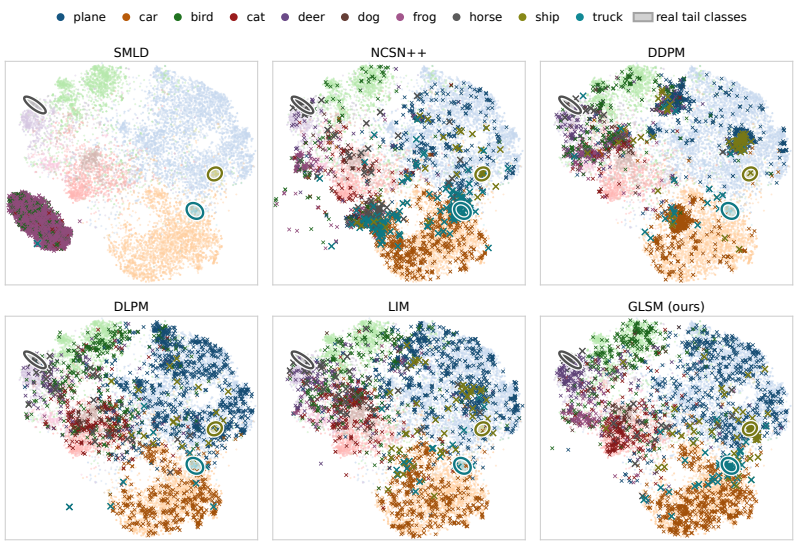


Figure 5: Feature-space visualization of CIFAR-10-LT samples at 128 NFEs. GLSM samples overlap broadly with the real-data support.

4.3 Long-Tailed Image Generation

We finally evaluate GLSM on CIFAR-10-LT, the class-imbalanced benchmark studied by LIM and DLPM (Yoon et al., 2023; Shariatian et al., 2025). We evaluate long-tailed generation through support coverage and class-conditional sample quality. Figure 5 visualizes the generated and real-data supports in Inception feature space, while Table 2 provides the corresponding quantitative comparison.

GLSM improves support coverage and class-conditional quality on the long-tailed distribution. Tail-3 coverage reaches 0.967, and GLSM obtains the highest recall and coverage in Table 2. After equalizing generated class counts, GLSM obtains the lowest class-wise FID in all ten classes, with a mean of 10.8 compared with 44.6 for LIM.

Method	Recall ↑	Coverage ↑	Tail-3 Coverage ↑	Avg. class FID ↓
GLSM (ours)	**0.687**	**0.868**	**0.967**	**10.8**
DLPM	0.603	0.779	0.629	48.5
LIM	0.602	0.659	0.549	44.6
NCSN++	0.596	0.552	0.845	59.4
DDPM	0.372	0.321	0.293	69.3

Table 2: CIFAR-10-LT generation near 128 NFEs.

5 Conclusion

We showed how to incorporate Lévy jump mechanisms into the sampling process for the generative models while retaining the standard Gaussian diffusion forward process. For an arbitrary prescribed Lévy measure, we construct the corresponding time-reversed Lévy-Ito SDE (Theorem 3.1) by introducing Lévy Langevin correctors (LLC) that enforce exact consistency of the marginals. To make the construction practical, we proposed DLSM to learn the Lévy score required by LLC using the same Gaussian clean–noisy training pairs as in standard denoising score matching.Building on the learned score, we further developed annealed pure-jump Lévy Langevin dynamics as a flexible sampling mechanism. Across controlled targets and image-generation benchmarks, our approach improves tail behavior and mode-mass recovery, enhances low-NFE sample quality, and increases support coverage and class-conditional fidelity under severe class imbalance at matched NFE budgets. These results position the generative jump law as an additional, user-controllable design dimension alongside the diffusion. Promising directions for future work include designing structured and data-adaptive Lévy measures, developing discretizations and solvers tailored to jump dynamics, and establishing sharper theory connecting jump-law choices to mixing, coverage, and sample quality.

Reproducibility Statement

We provide the assumptions and complete derivations for the theoretical results in Appendices A–B, including the reverse jump-diffusion construction, marginal preservation, and the corresponding learning objectives and conditional targets. Algorithm 1 specifies the annealed Lévy Langevin dynamics used for generation, and compound Poisson increment is specified in Appendix C. Appendix D provides the experimental setup and implementation details, including the construction of the synthetic targets and datasets, preprocessing, model architectures, Lévy measures, noise schedules, sampling updates, hyperparameter-selection procedures, baseline configurations, NFE accounting, and evaluation protocols. We report the sampling parameters used in the experiments and, where applicable, aggregate results over multiple independent generation seeds. Additional experiments in the appendix report sensitivity to the sampling configuration and the allocation of function evaluations across noise levels.

AI Use Statement

Generative AI tools were used to assist in the refinement of scientific writing. The authors reviewed and verified the AI-assisted outputs and take responsibility for the submitted content.

References

Krizhevsky Alex. Learning multiple layers of features from tiny images. *https://www. cs. toronto. edu/kriz/learning-features-2009-TR. pdf*, 2009.

Brian DO 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.

Arindam Banerjee, Xin Guo, and Hui Wang. On the optimality of conditional expectation as a bregman predictor. *IEEE Transactions on Information Theory*, 51(7):2664–2669, 2005.

Adrian Baule. Generative modelling with jump-diffusions. *Journal of Statistical Mechanics: Theory and Experiment*, 2026(5):054002, 2026.

Yu Cao, Jingrun Chen, Yixin Luo, and Xiang Zhou. Exploring the optimal choice for generative processes in diffusion models: Ordinary vs stochastic differential equations. *Advances in Neural Information Processing Systems*, 36:33420–33468, 2023.

Jacob Deasy, Nikola Simidjievski, and Pietro Liò. Heavy-tailed denoising score matching. *arXiv preprint arXiv:2112.09788*, 2021.

Weiye Gan, Tangjun Wang, and Zuoqiang Shi. Weighted laplacian flow: A deterministic particle flow with provable convergence. *arXiv preprint arXiv:2608.21831*, 2026.

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

Peter Holderrieth, Marton Havasi, Jason Yim, Neta Shaul, Itai Gat, Tommi Jaakkola, Brian Karrer, Ricky TQ Chen, and Yaron Lipman. Generator matching: Generative modeling with arbitrary markov processes. In *International Conference on Learning Representations*, volume 2025, pp. 52153–52219, 2025.

Lu-Jing Huang, Mateusz B Majka, and Jian Wang. Approximation of heavy-tailed distributions via stable-driven sdes. *Bernoulli*, 27(3):2040–2068, 2021.

Yuanfei Huang, Xiang Zhou, and Jinqiao Duan. Probability flow approach to the onsager–machlup functional for jump-diffusion processes. *SIAM Journal on Applied Mathematics*, 85(2):524–547, 2025.

Yuanfei Huang, Chengyu Liu, and Xiang Zhou. Lévy score function and score-based particle algorithm for nonlinear lévy–fokker–planck equations. *SIAM Journal on Numerical Analysis*, 64 (3):656–684, 2026.

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.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

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

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22 (4):730–751, 2025.

Eliya Nachmani, Robin San Roman, and Lior Wolf. Non gaussian denoising diffusion models. *arXiv preprint arXiv:2106.07582*, 2021.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International conference on machine learning*, pp. 7176–7185. PMLR, 2020.

David Oechsler. Lévy langevin monte carlo. *Statistics and Computing*, 34(1):37, 2024.

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.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11400–11410. IEEE, 2022.

Vadim Popov, Assel Yermekova, Tasnima Sadekova, Artem Khrapov, and Mikhail Kudinov. Improved sampling algorithms for lévy-itô diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 76407–76422, 2025.

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

Umut Şimşekli. Fractional langevin monte carlo: Exploring lévy driven stochastic differential equations for markov chain monte carlo. In *International Conference on Machine Learning*, pp. 3200–3209. PMLR, 2017.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

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

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

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

Laurens Van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of machine learning research*, 15(1):3221–3245, 2014.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

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.

A Lévy Score and Generative Dynamics

Section 3.1 uses the divergence representation of the adjoint jump operator to construct Brownian–Lévy dynamics with the Gaussian marginal path, while Section 3.3 uses the same representation to identify the invariant distribution of the Lévy Langevin process. We derive this representation first and then prove Theorems 3.1 and 3.3.

A.1 Divergence form of the adjoint jump operator

Let ν be a Lévy measure on $\mathbb{R}^d \setminus \{0\}$ satisfying $\int_{\mathbb{R}^d\setminus\{0\}} (1 \wedge \|r\|^2)\, \nu(\mathrm{d}r) < \infty$, and let χ be the truncation function used in the Lévy–Khintchine representation. For a smooth test function φ,

$$(\mathcal{L}_\nu \varphi)(x) = \int_{\mathbb{R}^d\setminus\{0\}} \left[\varphi(x+r) - \varphi(x) - \chi(r) r^\top \nabla\varphi(x)\right] \nu(\mathrm{d}r).$$

For finite Lévy measures, the compound-Poisson calculations below use the equivalent uncompensated convention $\chi \equiv 0$ together with the corresponding drift convention.

Let p be a positive density. We assume that the vector integral in Eq. (9) exists under the chosen truncation convention and that differentiation may be interchanged with the Lévy integral. The adjoint operator is then

$$\mathcal{L}_\nu^* p(x) = \int_{\mathbb{R}^d\setminus\{0\}} \left[p(x-r) - p(x) + \chi(r) r^\top \nabla p(x)\right] \nu(\mathrm{d}r).$$

Using

$$p(x-r) - p(x) = -\nabla \cdot \left(r \int_0^1 p(x-\eta r)\, \mathrm{d}\eta \right),$$

we obtain

$$\mathcal{L}_\nu^* p(x) = -\nabla \cdot \int_{\mathbb{R}^d\setminus\{0\}} r \left[\int_0^1 p(x-\eta r)\, \mathrm{d}\eta - \chi(r) p(x) \right] \nu(\mathrm{d}r).$$

The definition in Eq. (9) therefore gives $\mathcal{L}_\nu^* p = \nabla \cdot (p\, S_\nu[p])$. It also shows directly that $p\, S_\nu[p]$ is linear in p, which is the property used in Theorem 3.2.

For the isotropic stable normalization $\mathcal{L}_\alpha = -(-\Delta)^{\alpha/2}$,

$$\mathcal{L}_\alpha^* p = -(-\Delta)^{\alpha/2} p = \nabla \cdot \left[(-\Delta)^{(\alpha-2)/2} \nabla p \right].$$

Hence the Lévy score reduces, up to the normalization of the stable generator, to the fractional score in Eq. (7).

A.2 Proof of Theorem 3.1

For the time-dependent family $\{\nu_s\}_{0\le s\le T}$, assume that $s \mapsto \nu_s(B)$ is measurable for every Borel set B bounded away from the origin and that

$$\int_0^T \int_{\mathbb{R}^d\setminus\{0\}} (1 \wedge \|r\|^2)\, \nu_s(\mathrm{d}r)\, \mathrm{d}s < \infty.$$

We also assume that the integrals defining $S_{\nu_s}[q_s]$ exist and that the interchanges used in the preceding subsection hold for q_s. These are the integrability conditions referred to in Theorem 3.1.

The Gaussian forward SDE implies Eq. (12) for $q_s = p_{T-s}$. For the Brownian–Lévy process in Theorem 3.1, the Fokker–Planck equation has the form in Eq. (11). Substituting the drift from the theorem gives

$$b(\cdot, s) - \frac{h(s)^2}{2} S[q_s] - S_{\nu_s}[q_s] = -f(\cdot, T-s) + \frac{g(T-s)^2}{2} S[q_s].$$

Its forward equation is therefore exactly Eq. (12). Since $Y_0 \sim p_T = q_0$, uniqueness of the forward equation in the class specified by Theorem 3.1 yields $\mathrm{Law}(Y_s) = q_s = p_{T-s}$ for every $0 \le s \le T$. □

A.3 Proof of Theorem 3.3

Fix t and let $(P_\tau)_{\tau\ge 0}$ be the Markov semigroup generated by $\mathcal{A}_{\nu_t,p_t}$. The regularity conditions in Theorem 3.3 include conservativity of this semigroup and $p_t \in \mathcal{D}(\mathcal{A}^*_{\nu_t,p_t})$. By Eq. (9),

$$\mathcal{A}^*_{\nu_t,p_t} p_t = -\nabla \cdot (p_t S_{\nu_t}[p_t]) + \mathcal{L}^*_{\nu_t} p_t = 0.$$

Therefore

$$\frac{\mathrm{d}}{\mathrm{d}\tau} P^*_\tau p_t = P^*_\tau \mathcal{A}^*_{\nu_t,p_t} p_t = 0,$$

so $P^*_\tau p_t = p_t$ for every $\tau \ge 0$. Thus p_t is invariant. The ψ-irreducibility and Harris recurrence assumed in Theorem 3.3 make this invariant probability distribution unique, and the aperiodic skeleton assumption gives the stated total-variation convergence. □

B Denoising Lévy Score Matching

Section 3.2 reduces estimation of $S_{\nu_t}[p_t]$ to regression against the Lévy score of the Gaussian perturbation kernel. We first prove the denoising identity and then derive the conditional targets used for compound-Poisson and isotropic stable measures.

B.1 Proof of Theorem 3.2

Fix t and assume that $T_{\nu_t}(X_t, X_0; t)$ is integrable and that the Lévy integral in Eq. (9) may be interchanged with the mixture defining p_t. Since $p\, S_{\nu_t}[p]$ is linear in p,

$$p_t(x) S_{\nu_t}[p_t](x) = \int p_0(x_0) p_{0t}(x \mid x_0) T_{\nu_t}(x, x_0; t)\, \mathrm{d}x_0,$$

which is Eq. (14). Dividing by $p_t(x)$ turns the factor $p_0(x_0)p_{0t}(x \mid x_0)/p_t(x)$ into the conditional density of X_0 given $X_t = x$. Hence

$$\mathbb{E}[T_{\nu_t}(X_t, X_0; t) \mid X_t = x] = S_{\nu_t}[p_t](x)$$

for p_t-almost every x, proving Theorem 3.2. □

B.2 Compound-Poisson targets under Gaussian perturbations

Let $\nu_t(\mathrm{d}r) = \lambda_t \rho_t(\mathrm{d}r)$ and use the uncompensated convention $\chi \equiv 0$. With $X_t = \alpha_t X_0 + \sigma_t \varepsilon$ and $\varepsilon \sim \mathcal{N}(0, I)$, the Gaussian transition density satisfies

$$\frac{p_{0t}(X_t - \eta r \mid X_0)}{p_{0t}(X_t \mid X_0)} = \exp\left(\frac{\eta r^\top \varepsilon}{\sigma_t} - \frac{\eta^2 \|r\|^2}{2\sigma_t^2}\right).$$

Substitution into Eq. (9) gives Eq. (17).

The integrability assumption in Theorem 3.2 follows from a first moment of the jump-amplitude distribution. For fixed r and η,

$$\mathbb{E}_\varepsilon \exp\left(\frac{\eta r^\top \varepsilon}{\sigma_t} - \frac{\eta^2 \|r\|^2}{2\sigma_t^2}\right) = 1.$$

Tonelli's theorem and the triangle inequality then give

$$\mathbb{E}_\varepsilon \|T_{\nu_t}(\varepsilon, t)\| \le \lambda_t \int_{\mathbb{R}^d} \|r\| \, \rho_t(\mathrm{d}r).$$

Thus $T_{\nu_t}(\varepsilon, t)$ is integrable whenever ρ_t has a finite first moment.

Isotropic Gaussian jump amplitudes. Let $R \sim \mathcal{N}(0, \tau_t^2 I)$, $\kappa_t = \tau_t^2/\sigma_t^2$, and $z = \|\varepsilon\|^2/2$, as in Section 3.2. For fixed η, completing the square gives

$$\mathbb{E}_R\left[R \exp\left(\frac{\eta R^\top \varepsilon}{\sigma_t} - \frac{\eta^2 \|R\|^2}{2\sigma_t^2}\right)\right] = \frac{\eta \tau_t^2}{\sigma_t}(1+\kappa_t\eta^2)^{-d/2-1} \exp\left(\frac{\kappa_t \eta^2 z}{1+\kappa_t\eta^2}\right)\varepsilon.$$

Hence Eq. (17) becomes

$$T_{\nu_t}(\varepsilon, t) = -\frac{\lambda_t \tau_t^2}{\sigma_t}\varepsilon \int_0^1 \eta(1+\kappa_t\eta^2)^{-d/2-1} \exp\left(\frac{\kappa_t \eta^2 z}{1+\kappa_t\eta^2}\right) \mathrm{d}\eta.$$

Define

$$I_t(z) := \int_0^1 \eta(1+\kappa_t\eta^2)^{-d/2-1} \exp\left(\frac{\kappa_t \eta^2 z}{1+\kappa_t\eta^2}\right) \mathrm{d}\eta.$$

The substitution $y = z/(1+\kappa_t\eta^2)$ yields

$$I_t(z) = \frac{e^z z^{-d/2}}{2\kappa_t}\left[\gamma\left(\frac{d}{2}, z\right) - \gamma\left(\frac{d}{2}, \frac{z}{1+\kappa_t}\right)\right].$$

Since $\tau_t^2/(\sigma_t\kappa_t) = \sigma_t$, substitution gives Eq. (18).

B.3 Isotropic stable targets and Bregman regression

For the isotropic stable generator $\mathcal{L}_\alpha = -(-\Delta)^{\alpha/2}$, let $\phi(z) = (2\pi)^{-d/2} e^{-\|z\|^2/2}$. Set $a = (d+\alpha)/2$, $b = d/2+1$, and $c_{\alpha,d} = 2^{\alpha/2-1}\Gamma(a)/\Gamma(b)$. Fourier inversion gives

$$\mathcal{F}[p\, S_\alpha[p]](\xi) = i\xi \, \|\xi\|^{\alpha-2} \mathcal{F}[p](\xi),$$

so for the standard Gaussian density,

$$\phi(z) S_\alpha[\phi](z) = -(2\pi)^{-d/2} c_{\alpha,d}\, z \, {}_1F_1\left(a; b; -\frac{\|z\|^2}{2}\right).$$

For $p_{0t}(x \mid x_0) = \sigma_t^{-d}\phi((x-\alpha_t x_0)/\sigma_t)$ and $\varepsilon = (x-\alpha_t x_0)/\sigma_t$, scaling of the fractional operator gives

$$T_\alpha(\varepsilon, t) = -c_{\alpha,d}\sigma_t^{1-\alpha}\varepsilon e^{\|\varepsilon\|^2/2} {}_1F_1\left(a; b; -\frac{\|\varepsilon\|^2}{2}\right).$$

This is the explicit Gaussian conditional target referred to in Section 3.2.

For $1 < \alpha < 2$, we have $b - a = 1 - \alpha/2 > 0$. The large-negative-argument asymptotic of the Kummer function gives, with $r = \|\varepsilon\|$,

$${}_1F_1\left(a; b; -\frac{r^2}{2}\right) \asymp C_{a,b} r^{-d-\alpha}, \qquad r \to \infty.$$

Therefore

$$\|T_\alpha(\varepsilon, t)\| \asymp C_{\alpha,d,t} e^{r^2/2} r^{1-d-\alpha}.$$

Near the origin, $T_\alpha(\varepsilon, t) = O(r)$. Under $\varepsilon \sim \mathcal{N}(0, I)$, the radial integral for the first moment behaves as $\int^\infty r^{-\alpha}\,\mathrm{d}r$, whereas the second moment retains an $e^{r^2/2}$ factor. Hence $\mathbb{E}\|T_\alpha\| < \infty$ and $\mathbb{E}\|T_\alpha\|^2 = \infty$ for $1 < \alpha < 2$, as stated in Section 3.2.

The finite first moment is sufficient for the Bregman objective used there. Let T be an integrable vector target, let X be the conditioning variable, and write $m(X) = \mathbb{E}[T \mid X]$. For $\Psi_\delta(v) = \delta^2(\sqrt{1 + \|v\|^2/\delta^2} - 1)$, the associated Bregman divergence satisfies

$$\mathbb{E}[D_{\Psi_\delta}(T, z) \mid X] = \mathbb{E}[\Psi_\delta(T) \mid X] - \Psi_\delta(m(X)) + D_{\Psi_\delta}(m(X), z).$$

Strict convexity makes $m(X)$ the unique minimizer of the conditional risk (Banerjee et al., 2005). The bounds $\|\nabla\Psi_\delta(z)\| \le \delta$ and $\Psi_\delta(y) \le \delta\|y\|$ further imply

$$0 \le D_{\Psi_\delta}(y, z) \le 2\delta\|y\| + \delta^2.$$

Thus $\mathbb{E}\|T\| < \infty$ is sufficient for finite population risk. Applying this result to the stable conditional target proves the claims following Theorem 3.2.

C Compound-Poisson Increments

Algorithm 1 requires an increment of the Lévy process with measure ν_{t_i} over one numerical step. For $\nu_{t_i}(\mathrm{d}r) = \lambda_i\rho_i(\mathrm{d}r)$ under the uncompensated convention, draw $N_i \sim \mathrm{Poisson}(\lambda_i\eta_i)$. Conditional on N_i, draw $R_{i,1}, \dots, R_{i,N_i} \overset{\text{iid}}{\sim} \rho_i$. Then

$$\Delta L_{\eta_i}^{\nu_{t_i}} = \sum_{k=1}^{N_i} R_{i,k}.$$

Substituting this increment into the update in Algorithm 1 gives

$$X \leftarrow X + \eta_i S_\theta^\nu(X, t_i) + \sum_{k=1}^{N_i} R_{i,k}.$$

Thus Algorithm 1 samples jumps from the same measure ν_{t_i} that defines the score $S_{\nu_{t_i}}[p_{t_i}]$.

D Additional Experimental Details and Results

This appendix provides the implementation and evaluation details for Section 4, together with supplementary results that expand the controlled, low-NFE image, and CIFAR-10-LT comparisons.

D.1 Controlled Distributions

D.1.1 Targets and Training

Truncated Cauchy target. Following Gan et al. (2026), we use independent Cauchy coordinates truncated to $[-15, 15]$, with joint density

$$p_{\mathrm{C}}(x) = \prod_{j=1}^{2} \frac{\mathbf{1}_{\{|x_j| \le 15\}}}{2\arctan(15)(1 + x_j^2)}.$$

Training and sampling are performed in coordinates $z = x/15$; generated samples are mapped back to the original coordinates before evaluation and are not clipped to the target support.

Imbalanced Gaussian mixture. We use the nine-component mixture adapted from Shariatian et al. (2025). Its means are (i, j) for $i, j \in \{0, 1, 2\}$, with common covariance $0.1^2 I_2$. In row-major order, with i as the outer index, the mixture weights are

$$(\pi_1, \dots, \pi_9) = (0.01, 0.10, 0.30, 0.20, 0.02, 0.15, 0.02, 0.15, 0.05).$$

We use model coordinates $z = x - (1, 1)$ and evaluate in the original coordinates. Independent training, validation, and test sets contain 200,000, 100,000, and 200,000 samples for each target. All methods use the same cached splits.

Table 3: Synthetic sampling settings for Table 1. The three η columns correspond to SMLD, GLSM-CP, and GLSM-α.

Target	NFE	$\sigma_{\max}$	$\sigma_{\min}$	a_0	α	η_{B}	η_{CP}	η_α
Cauchy	32	0.3	10^{-3}	8	1.5	1.25	1.25	0.4942
GMM	128	1.5	10^{-2}	4	1.7	0.7861	0.4942	0.4942

Synthetic score models. The Brownian-score and Lévy-score models use separate noise-conditioned residual multilayer perceptrons with width 256, four residual blocks, and 830,723 parameters each. Both use Fourier embeddings of the log-noise level and the output parameterization $S_\theta = v_\theta \exp(\mathrm{clip}(m_\theta, -25, 25))$. The Brownian model predicts $S^{\mathrm{B}}_\theta(x,\sigma) = \sigma \widehat{s}_{\mathrm{B}}(x,\sigma)$ and is trained by denoising score matching (Vincent, 2011). For stable index α, the Lévy-score model predicts $S^{\alpha}_\theta(x,\sigma) = \sigma^{\alpha-1}\widehat{s}_\alpha(x,\sigma)$. Both models use Gaussian perturbations $X_\sigma = X_0 + \sigma\varepsilon$, with $\varepsilon \sim \mathcal{N}(0, I_2)$ and $\log\sigma$ sampled uniformly over $[\log(\sigma_{\min}/2), \log 4.5]$.

Stable DLSM objective. For each $\alpha \in \{1.5, 1.7, 1.8, 1.9\}$, we train a separate model using the Gaussian conditional stable target derived in Appendix B. Let $\widetilde{T}_\alpha = \sigma^{\alpha-1} T_\alpha$ denote the rescaled target. We minimize

$$\mathbb{E}\Big[D_{\Psi_3}\Big(\widetilde{T}_\alpha, S^\alpha_\theta(X_\sigma,\sigma)\Big)\Big], \qquad \Psi_3(v) = 9\left(\sqrt{1+\|v\|^2/9} - 1\right).$$

The implementation omits the target-only term $\Psi_3(\widetilde{T}_\alpha)$ during optimization and evaluates the target coefficient in log coordinates, with an upper exponent guard of 300 to avoid overflow.

D.1.2 Sampling

Synthetic samplers start from $\mathcal{N}(0, \sigma^2_{\max} I_2)$ and traverse a decreasing geometric noise schedule with one update at each selected level. The Cauchy experiments use $(\sigma_{\max}, \sigma_{\min}) = (0.3, 10^{-3})$, and the Gaussian-mixture experiments use $(1.5, 10^{-2})$.

Compound-Poisson LLC. At noise level σ, Gaussian jump amplitudes have variance $v_\sigma = a_0^2\sigma^2/d$ and intensity $\lambda_\sigma = 2/v_\sigma$. A jump has root-mean-square length $a_0\sigma$, and the accumulated jump covariance per unit time is $2I_d$. The synthetic update is

$$X^+ = X + \Pi_{10\eta\sigma}(h\widehat{s}_{\nu_\sigma}(X)) + \sqrt{N v_\sigma}\, Z, \qquad h = (\eta\sigma)^2, \tag{21}$$

where $N \sim \mathrm{Poisson}(2h/v_\sigma)$ and $Z \sim \mathcal{N}(0, I_d)$ are independent, $\Pi_R(z) = z\min\{1, R/\|z\|\}$ for $z \neq 0$, and $\Pi_R(0) = 0$. Conditional on N, the Gaussian jump sum has covariance $N v_\sigma I_d$, so the compound-Poisson increment is sampled exactly. Synthetic SMLD uses the same drift cap, with $\widehat{s}_{\mathrm{B}}$ replacing $\widehat{s}_{\nu_\sigma}$ and $\sqrt{2h}Z$ replacing the jump increment. Neither synthetic sampler applies final denoising.

Stable LLC. The stable sampler uses $h = (\eta\sigma)^\alpha$ and the learned field $\widehat{s}_\alpha = \sigma^{1-\alpha} S^\alpha_\theta$. If $U_{\alpha/2}$ is a positive stable variable with Laplace transform $\mathbb{E}e^{-tU_{\alpha/2}} = e^{-t^{\alpha/2}}$, the implemented update is

$$X^+ = X + \Pi_{10\eta\sigma}(h\widehat{s}_\alpha(X,\sigma)) + h^{1/\alpha}\sqrt{2\min\{U_{\alpha/2}, 10\}}\, Z.$$

The positive-stable scale-mixture representation gives an isotropic stable increment with characteristic function $\exp(-h\|\xi\|^\alpha)$. We cap $U_{\alpha/2}$ at 10 in the numerical sampler and keep this value fixed across budgets and random seeds.

Parameter selection. The step-size parameter η is selected separately for each method and budget by validation sliced W_1. The compound-Poisson scale is selected from $a_0 \in \{4, 6, 8\}$, and the stable index from $\alpha \in \{1.5, 1.7, 1.8, 1.9\}$ by averaging validation log-sliced-W_1 over budgets $16, 32, 64, 128, 256$. The selected a_0 and α are then held fixed across budgets. Table 3 gives the settings used for the main-text operating points.

D.1.3 Evaluation

Each synthetic operating point uses 50,000 generated samples for each of ten seeds, 101–110, and is evaluated against the independent test split. Metrics are computed separately for each seed after undoing the model-space transformation. Sliced W_1 is approximated with 256 uniformly sampled directions θ_k on the unit circle:

$$\widehat{\mathrm{SW}}_1(\widehat{P},\widehat{Q}) = \frac{1}{256}\sum_{k=1}^{256} W_1\left((\theta_k^\top)_\#\widehat{P},(\theta_k^\top)_\#\widehat{Q}\right).$$

The one-dimensional distances are computed by quantile interpolation on a common grid. At most 100,000 reference points are used, and projection and reference-subsampling seeds are shared across methods for each evaluation seed.

For the truncated Cauchy target, the exact quantile function of $|X_j|$ is $Q(u) = \tan(u \arctan 15)$. For 512 equally spaced values u_k from 0.95 to 0.9995, we compute

$$\mathrm{MSLE}_{\mathrm{tail}} = \frac{1}{2\cdot 512}\sum_{j=1}^{2}\sum_{k=1}^{512}\left[\log\max\{\widehat{Q}_j(u_k), 10^{-12}\} - \log Q(u_k)\right]^2.$$

For the Gaussian mixture, each generated point is assigned to its nearest component mean. If $\widehat{\pi}_k$ is the assignment frequency, the mode-mass error is $\mathrm{TV}_{\mathrm{mode}} = \frac{1}{2}\sum_{k=1}^{9}|\widehat{\pi}_k - \pi_k|$.

D.2 Image Generation

D.2.1 Data and DLSM Training

CIFAR-10 uses the standard training set (Alex, 2009). CIFAR-10-LT retains $\mathrm{round}(5000\cdot 100^{-c/9})$ images from class $c \in \{0,\dots,9\}$, using seed 0 for within-class subsampling. The resulting dataset contains 12,408 images and has a largest-to-smallest class-size ratio of 100. NCSN++ models use RGB pixel values in $[0,1]$.

CelebA-64 uses the aligned CelebA images (Liu et al., 2015) with the preprocessing of the released DDIM CelebA model (Song et al., 2021a): a 128×128 crop with upper-left pixel $(25,57)$ followed by bicubic resizing to 64×64. The score model uses pixels in $[-1,1]$. Generated outputs are converted to clipped, rounded, 8-bit RGB images before evaluation.

For CIFAR-10 and CIFAR-10-LT, we use the discrete-VE NCSN++ architecture of Song et al. (2021b) with 62.7M parameters. For CelebA-64, we use the 64×64 U-Net architecture of Song et al. (2021a) with base width 64, channel multipliers $(1,2,2,4)$, two residual blocks per resolution, attention at resolution 16, and 17.2M parameters.

All image models use Gaussian perturbations $X_\sigma = X_0 + \sigma\varepsilon$, with $\varepsilon\sim\mathcal{N}(0,I_d)$. For the compound-Poisson corrector, the Gaussian jump amplitudes satisfy $\rho_\sigma = \mathcal{N}(0,v_\sigma I_d)$ with $v_\sigma = a_0^2\sigma^2/d$, $a_0 = 8$, and intensity $\lambda_\sigma = 2/v_\sigma$. For CIFAR-10 and CIFAR-10-LT, the noise index is sampled uniformly from the 1000-level geometric grid between $\sigma_{\max} = 50$ and $\sigma_{\min} = 0.01$; for CelebA-64, $\log\sigma$ is sampled uniformly over $[\log 0.005, \log 50]$.

D.2.2 Sampling and Baselines

Image baselines include SMLD and NCSN++ (Song & Ermon, 2019; Song et al., 2021b), DDPM (Ho et al., 2020), LIM (Yoon et al., 2023), and DLPM (Shariatian et al., 2025). DDPM uses ancestral sampling, LIM uses its stochastic SDE-A sampler (Popov et al., 2025) with clipping parameter 20, and DLPM uses stable index $\alpha = 1.8$. On CIFAR-10-LT, the retained checkpoints are DDPM at 150,000 updates, DLPM at epoch 500, and LIM at epoch 700. The CelebA-64 LIM and DLPM models are trained on the same preprocessed images for 100,000 updates with batch size 128.

Image sampling selects uniformly spaced indices from the 1000-level geometric grid between 50 and 0.01. We use one LLC update at each selected level, so the number of selected levels equals the LLC NFE. With $a_0 = 8$ and $h = 2(\eta\sigma)^2$, the image update is

$$X^+ = X + \mathcal{T}_{\sqrt{2hd}}(h\widehat{s}_{\nu_\sigma}(X,\sigma)) + \sqrt{Nv_\sigma}\,Z, \qquad N\sim\mathrm{Poisson}(2h/v_\sigma), \qquad Z\sim\mathcal{N}(0,I_d), \quad (22)$$

Table 4: GLSM image sampling settings. Each entry reports NFE/η.

Dataset	NFE / η			
CIFAR-10	32/0.2263	64/0.16	128/0.32	256/0.2263
CIFAR-10-LT	32/0.48	64/0.16	128/0.32	256/0.2263
CelebA-64	32/0.2263	64/0.32	128/0.16	256/0.1131

Table 5: CIFAR-10 validation FID with one update per stage and fixed noise endpoints.

Stages L	SMLD	GLSM (ours)
32	323.8	92.7
64	312.6	45.9
128	269.3	22.3
256	202.9	18.6
512	117.6	17.4
1024	66.7	17.0

where $\mathcal{T}_R(z) = z/(1 + \|z\|/R)$. The final state at the smallest noise level is returned directly.

Image sampler settings are selected by validation FID computed from 2048 generated samples with seed 0; CIFAR-10-LT uses the long-tailed training reference. Table 4 gives the GLSM step-size parameters at the main reporting budgets.

Image SMLD applies $X^+ = X + h\widehat{s}_{\mathrm{B}}(X, \sigma) + \sqrt{2h}Z$ with $h = 2(\eta\sigma)^2$. NCSN++ is implemented by predictor-corrector (PC) sampler, which alternates Langevin corrector updates with a reverse-diffusion predictor (Song et al., 2021b); its corrector step size is selected through the sampler signal-to-noise ratio. Each baseline is tuned within its native parameterization.

D.2.3 Image Evaluation

We compute FID (Heusel et al., 2017) from 2048-dimensional Inception pool features using Clean-FID in `legacy_tensorflow` mode (Parmar et al., 2022). For empirical feature means $\mu_{\mathrm{G}}, \mu_{\mathrm{R}}$ and covariance matrices $C_{\mathrm{G}}, C_{\mathrm{R}}$,

$$\mathrm{FID} = \|\mu_{\mathrm{G}} - \mu_{\mathrm{R}}\|^2 + \mathrm{tr}\Big(C_{\mathrm{G}} + C_{\mathrm{R}} - 2(C_{\mathrm{R}}^{1/2} C_{\mathrm{G}} C_{\mathrm{R}}^{1/2})^{1/2}\Big).$$

The feature pipeline includes bilinear resizing to 299×299. The references for the reported CIFAR-10 and CelebA-64 FIDs contain the 50,000 CIFAR-10 training images and the 197,599 preprocessed CelebA-64 training images, respectively. Each image run contains 10,000 generated samples. CIFAR-10 and CIFAR-10-LT use three generation seeds; CelebA-64 contains both one-seed and three-seed operating points. Means and sample standard deviations are reported whenever multiple seeds are available.

D.3 Robustness to the Annealing Schedule

We examine how generation quality changes with the number of intermediate noise levels, the updates performed at each level, and the terminal noise scale. The image experiments use compound-Poisson jumps with scale $a_0 = 8$. We use L sampling stages and M updates per stage in the sampling process.

D.3.1 Coarse noise ladders

We first fix $(\sigma_{\max}, \sigma_{\min}) = (50, 0.01)$ and use one update per stage, the results are summarized in Table 5.

GLSM reaches validation FID below 70 with 64 distinct noise levels, while SMLD first reaches this threshold at 1024 stages. Thus GLSM uses approximately 16 times fewer distinct levels at this threshold on the tested grid.

Table 6: CIFAR-10 validation FID on the same 32-level ladder.

Updates per level M	SMLD	GLSM (ours)
1	323.8	92.7
2	312.0	134.0
4	298.2	74.3
8	229.5	35.6
16	153.5	25.1
32	97.1	20.6
64	59.8	—

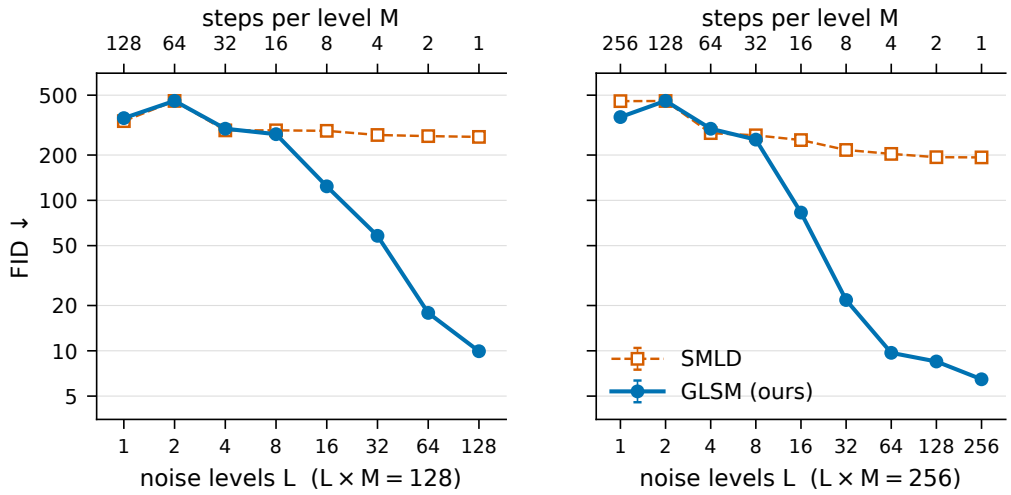


Figure 6: Fixed-budget allocation on CIFAR-10.

D.3.2 Within-level updates and local relaxation

To separate ladder resolution from within-level computation, we fix $L = 32$ and vary M. Table 6 shows that additional updates improve both methods over much of the tested range. GLSM obtains FID 92.7 with one update per level, while SMLD first reaches FID below 100 with 32 updates per level. The GLSM result is non-monotone at small M: increasing M from 1 to 2 worsens FID to 134.0.

D.3.3 Fixed-budget allocation across noise levels

The preceding experiments vary either the number of noise levels or the number of updates per level while leaving the other quantity fixed. We further compare the two allocations at a matched sampling budget. Specifically, we fix the product LM and vary the split between the number of distinct noise levels L and the number of updates M performed at each level. Figure 6 reports the resulting FID on CIFAR-10, while Figure 7 reports the corresponding FID against the long-tailed reference on CIFAR-10-LT.

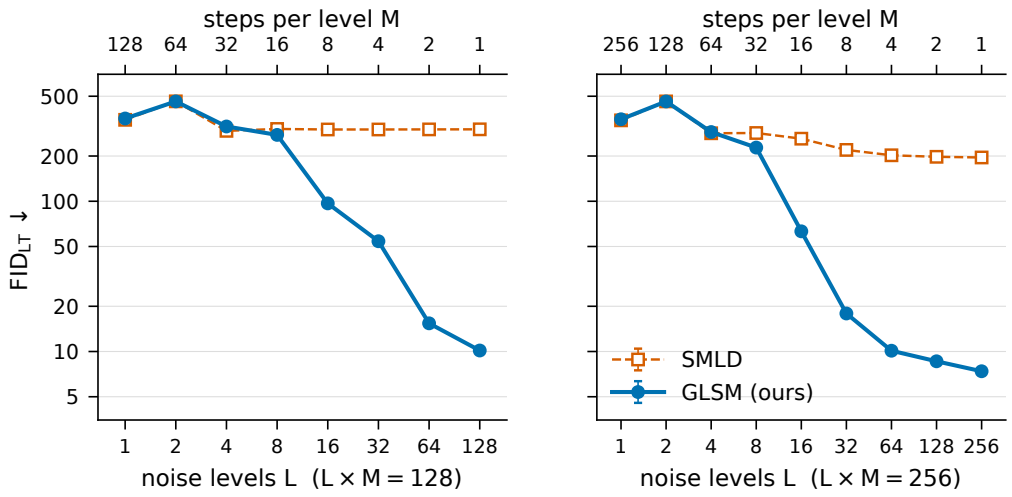


Figure 7: Fixed-budget allocation on CIFAR-10-LT.

Table 7: CIFAR-10 terminal-noise diagnostics.

		$\sigma_{\max}$			
Method	NFE	6.19	12.46	24.85	50.00
SMLD	128	254.3	253.1	297.6	269.3
SMLD	256	166.4	183.3	197.7	202.9
SMLD	512	94.6	100.4	109.5	117.6
NCSN++	128	60.5	54.8	61.5	71.4
NCSN++	256	30.7	23.9	23.8	24.8
NCSN++	512	22.4	18.3	18.3	18.3
GLSM (ours)	128	25.9	21.5	21.9	22.3
GLSM (ours)	256	22.7	18.8	18.6	18.6
GLSM (ours)	512	18.0	17.5	17.7	17.4

Across both datasets and both budgets, GLSM reaches useful FID values with substantially fewer distinct noise levels than SMLD. For example, with 128 NFEs, GLSM reaches FID 58.1 on CIFAR-10 and 54.2 on CIFAR-10-LT using only $L = 32$ levels and $M = 4$ updates per level, whereas SMLD remains at FID 271.9 and 300.4, respectively. With $LM = 256$, the corresponding $(L, M) = (32, 8)$ configurations obtain FID 21.8 and 17.9 for GLSM, compared with 216.2 and 219.6 for SMLD. The comparison also shows that, at fixed total update budget, both samplers benefit from allocating more computation to distinct noise levels rather than repeatedly updating at a small number of levels. Thus the relevant conclusion is not that GLSM prefers coarse ladders, but that it remains effective under substantially coarser discretizations of the annealing path.

D.3.4 Terminal noise and the forward horizon

We vary $\sigma_{\max}$ while keeping $\sigma_{\min} = 0.01$. Under a fixed increasing noise schedule, $\sigma_{\max} = \sigma(T)$, so a larger terminal scale corresponds to a longer forward noising horizon. For each terminal scale and backbone-evaluation budget C, we select the number of stages, updates per stage, and step-size parameter using validation FID.

The prior diagnostic becomes less discriminative as $\sigma_{\max}$ increases, consistent with reduced terminal mismatch. For terminal scales 12.46, 24.85, and 50, GLSM's validation FID ranges across the rounded entries are 0.8, 0.2, and 0.3 at NFEs 128, 256, and 512, respectively. SMLD benefits from a shorter horizon at NFEs 256 and 512. NCSN++ also benefits at NFE 128, while becoming comparatively insensitive to terminal noise at the larger budgets. Thus the horizon sensitivity depends on both the sampler and the computational budget.

Table 8: Truncated Cauchy sliced W_1 after 128 NFEs.

$\sigma_{\max}$	Selected a_0	SMLD	GLSM (ours)
0.1	6	0.1230 ± 0.0036	0.0544 ± 0.0042
0.2	8	0.2970 ± 0.0075	0.0502 ± 0.0034
0.3	8	0.3170 ± 0.0084	0.0527 ± 0.0041
0.5	8	0.3440 ± 0.0087	0.0551 ± 0.0036
1.5	8	0.3960 ± 0.0079	0.0614 ± 0.0042

Truncated Cauchy. We also examine the truncated Cauchy target from Appendix D.1. Table 8 compares methods at 128 sampler updates and gives their evaluation counts separately. The jump scale is selected by validation from $a_0 \in \{4, 6, 8\}$ for each terminal scale and then held fixed across sampling budgets.

GLSM is less sensitive to terminal noise in this sweep: its mean sliced W_1 ranges from 0.0502 to 0.0614, while SMLD's increases from 0.1230 to 0.3960. This behavior is estimator dependent. A longer horizon can therefore improve a Brownian baseline and reduce the relative advantage of GLSM.

D.4 Additional CIFAR-10-LT Results

The supplementary CIFAR-10-LT results expand the two quantities reported in Section 4.3: support coverage and generation quality within individual classes.

D.4.1 Evaluation Protocol

We use feature-space recall (Kynkäänniemi et al., 2019) and coverage (Naeem et al., 2020) with $k = 5$. Let $\mathcal{R}$ denote the Inception features of all $12{,}408$ real CIFAR-10-LT images, $\mathcal{G}$ the $10{,}000$ generated features from one seed, and $\mathcal{R}_c \subset \mathcal{R}$ the real images with true label c. Write $r_k(z; \mathcal{S})$ for the distance from $z \in \mathcal{S}$ to its kth nearest other point in $\mathcal{S}$. We compute

$$\text{Recall}_c = \frac{1}{|\mathcal{R}_c|} \sum_{x \in \mathcal{R}_c} \mathbf{1}\{\exists y \in \mathcal{G} : \|x - y\| \le r_k(y; \mathcal{G})\},$$

$$\text{Coverage}_c = \frac{1}{|\mathcal{R}_c|} \sum_{x \in \mathcal{R}_c} \mathbf{1}\left\{\min_{y \in \mathcal{G}} \|x - y\| \le r_k(x; \mathcal{R})\right\}.$$

Overall values weight the class-wise quantities by the class frequencies of the long-tailed reference. Tail-3 values are unweighted averages over horse, ship, and truck. Reported values are averaged over three generation seeds.

For class-wise FID, let $\mathcal{G}_{m,c}$ contain samples from method m assigned to class c by the frozen classifier, pooled across the three generation seeds, and let $\mathcal{R}_c^{\text{bal}}$ contain the 5000 balanced CIFAR-10 training images with true label c. We equalize generated sample counts through

$$n_c = \min\left\{1000, \min_{m \in \mathcal{M}} |\mathcal{G}_{m,c}|\right\}, \qquad \mathcal{M} = \{\text{GLSM}, \text{NCSN++}, \text{DDPM}, \text{DLPM}, \text{LIM}\}.$$

For each method and class, we draw n_c generated samples without replacement and compute FID against $\mathcal{R}_c^{\text{bal}}$, averaging over five draws. Applying the same procedure to n_c held-out real images gives a class-specific real-versus-real reference $\text{FID}_c^{\text{floor}}$. We report

$$\widetilde{\text{FID}}_{m,c} = \frac{1}{5} \sum_{r=1}^{5} \text{FID}\left(\mathcal{G}_{m,c}^{(r,n_c)}, \mathcal{R}_c^{\text{bal}}\right) - \text{FID}_c^{\text{floor}}.$$

Generated sample counts are matched across methods within each class, and the same real-versus-real reference is subtracted from every method in that class.

Figure 5 uses the matched 128-NFE sample pools. We jointly embed Inception features of the real CIFAR-10-LT reference and generated samples, reduce the combined features to 50 dimensions by PCA, and apply two-dimensional Barnes–Hut t-SNE with perplexity 30, PCA initialization, random seed 0, and 1000 iterations (Van der Maaten & Hinton, 2008; Van der Maaten, 2014).

Table 9: CIFAR-10-LT coverage at low NFE budgets.

Method	NFE	Coverage ↑
GLSM (ours)	128	**0.868**
DDIM (Song et al., 2021a)	100	0.835
DPM-Solver (Lu et al., 2022)	100	0.826
DPM-Solver++ (Lu et al., 2025)	100	0.809
LIM-SDE-E (Popov et al., 2025)	100	0.787

D.4.2 Additional Support-Coverage Comparisons

The broader low-NFE comparison uses DDIM, DPM-Solver, and DPM-Solver++ with the DDPM checkpoint retained at 150,000 updates, and LIM-SDE-E with the LIM checkpoint retained at epoch 700. DDIM uses $\eta = 0$ on a quadratic time grid. DPM-Solver and DPM-Solver++ use multistep order 3 at the reported budget; DPM-Solver uses a log-SNR grid with a final denoising evaluation, while DPM-Solver++ uses a uniform-time grid without an additional denoising evaluation (Lu et al., 2022; 2025). LIM-SDE-E follows the exact-marginal-form update of Popov et al. (2025) with $\eta_t = t^{1/5}$ on the published time grid.

Table 10 expands the matched comparison in the main text to include both overall and Tail-3 recall and coverage.

Table 10: CIFAR-10-LT support recovery near 128 NFEs. Tail-3 averages horse, ship, and truck.

Method	Recall ↑	Coverage ↑	Tail-3 Recall ↑	Tail-3 Coverage ↑
GLSM (ours)	**0.687**	**0.868**	**0.758**	**0.967**
SMLD	0.000	0.007	0.000	0.001
NCSN++	0.596	0.552	0.677	0.845
DDPM	0.372	0.321	0.442	0.293
DLPM	0.603	0.779	0.617	0.629
LIM	0.602	0.659	0.619	0.549

At the larger budget, the same coverage pattern persists (Table 11).

Table 11: CIFAR-10-LT coverage near 256 NFEs. Tail-3 averages horse, ship, and truck.

Method	Coverage ↑	Tail-3 Coverage ↑
GLSM (ours)	**0.905**	**0.977**
NCSN++	0.858	0.970
DLPM	0.787	0.636
LIM	0.745	0.637
DDPM	0.695	0.613

D.4.3 Class-Conditional Quality

After equalizing generated class counts, GLSM gives the lowest class-wise FID in all ten classes. Table 12 reports the complete class-wise comparison.

The equal-count analysis also gives the highest mean class-wise recall for GLSM.

	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Mean
GLSM (ours)	**4.6**	**3.1**	**7.8**	**9.4**	**5.6**	**25.3**	**9.8**	**12.2**	**13.8**	**16.1**	**10.8**
DLPM	9.7	4.3	26.0	29.9	30.1	57.0	74.8	69.3	51.3	133.1	48.5
LIM	24.7	13.2	31.1	37.2	30.1	57.9	43.6	76.0	65.2	67.3	44.6
NCSN++	49.7	26.4	80.7	61.2	42.0	72.5	78.8	53.8	57.8	71.0	59.4
DDPM	63.9	47.9	51.5	65.2	50.1	73.4	63.7	75.3	88.7	113.9	69.3

Table 12: Class-wise FID on CIFAR-10-LT near 128 NFEs. Lower is better.

Table 13: Per-class recall near 128 NFEs after equalizing generated class counts across methods. The final column is the unweighted class mean.

Method	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Mean
GLSM (ours)	0.708	0.652	0.695	0.680	0.722	0.781	0.714	0.781	0.743	0.761	**0.724**
NCSN++	0.629	0.614	0.637	0.556	0.658	0.730	0.639	0.782	0.644	0.716	0.661
DDPM	0.379	0.406	0.513	0.364	0.575	0.475	0.511	0.606	0.556	0.621	0.501
DLPM	0.645	0.580	0.671	0.552	0.717	0.609	0.591	0.815	0.651	0.808	0.664
LIM	0.617	0.581	0.699	0.555	0.735	0.659	0.642	0.834	0.717	0.793	0.683

D.5 Additional Generated Samples

D.5.1 Unconditional samples at matched sampling budgets

Figure 8 shows unconditional CIFAR-10-LT samples at approximately 128 NFEs. For each method, we display the first twelve samples from the stored seed-1 generation pool, without class-based or visual selection.

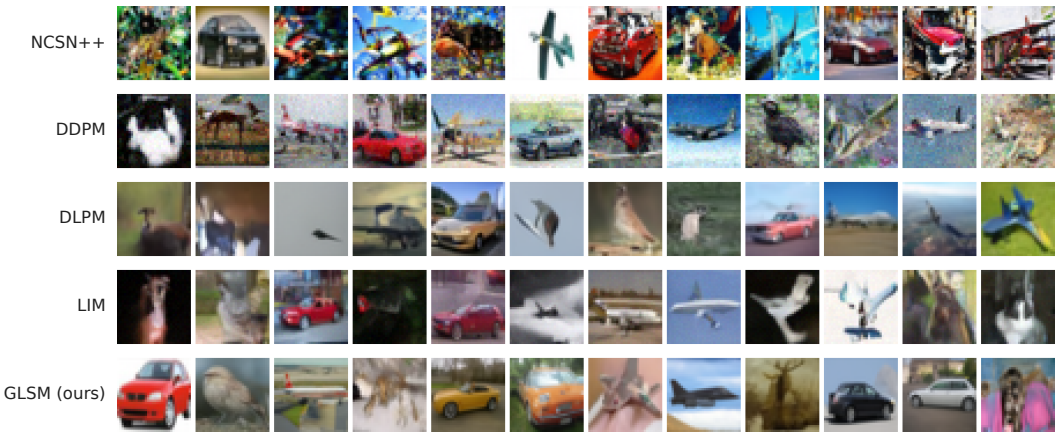


Figure 8: Unconditional CIFAR-10-LT samples at near 128 NFEs.

Figure 9 gives the corresponding comparison on balanced CIFAR-10.

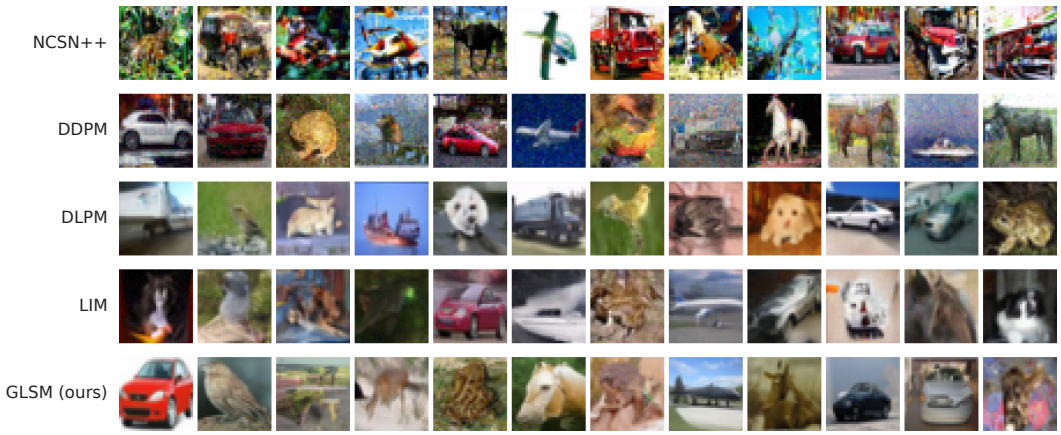


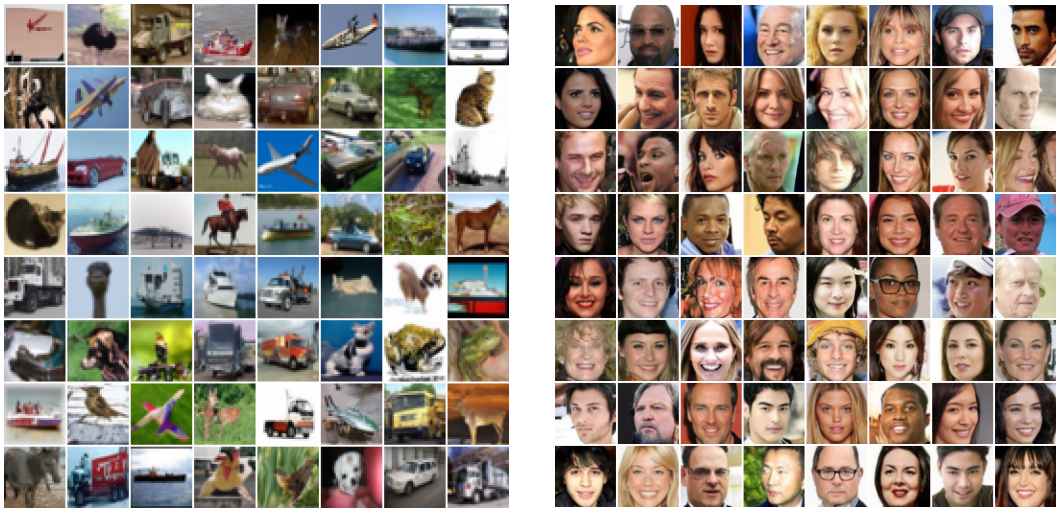
Figure 9: Unconditional CIFAR-10 samples at near 128 NFEs.

D.5.2 Rare-class samples on CIFAR-10-LT



Figure 10: Generation of rare CIFAR-10-LT classes at near 128 NFEs.

D.5.3 Additional high-budget GLSM samples



(a) CIFAR-10 (b) CelebA-64

Figure 11: Additional samples generated by GLSM on CIFAR-10 and CelebA-64.