DENOISING LÉVY PROBABILISTIC MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Exploring noise distributions beyond Gaussian in diffusion models remains an open challenge. While Gaussian-based models succeed within a unified SDE framework, recent studies suggest that heavy-tailed noise distributions, like α-stable distributions, may better handle mode collapse and effectively manage datasets exhibiting class imbalance, heavy tails, or prominent outliers. Recently, Yoon et al. (NeurIPS 2023), presented the Lévy-Itô model (LIM), directly extending the SDE-based framework to a class of heavy-tailed SDEs, where the injected noise followed an α-stable distribution, a rich class of heavy-tailed distributions. However, the LIM framework relies on highly involved mathematical techniques with limited flexibility, potentially hindering broader adoption and further development. In this study, instead of starting from the SDE formulation, we extend the denoising diffusion probabilistic model (DDPM) by replacing the Gaussian noise with α-stable noise. By using only elementary proof techniques, the proposed approach, *Denoising Lévy Probabilistic Model* (DLPM), boils down to vanilla DDPM with minor modifications. As opposed to the Gaussian case, DLPM and LIM yield different training algorithms and different backward processes, leading to distinct sampling algorithms. These fundamental differences translate favorably for DLPM as compared to LIM: our experiments show improvements in coverage of data distribution tails, better robustness to unbalanced datasets, and improved computation times requiring smaller number of backward steps.

1 INTRODUCTION

The evolution of generative models has introduced several approaches, with diffusion models emerging as one of the most prominent. These models transform a data distribution into a Gaussian distribution via a forward noising process and then learn to reverse it. The foundational work on denoising in this context was presented by Sohl-Dickstein et al. (2015), where the goal is to reverse a Markov chain that progressively adds Gaussian noise to the data. This framework culminated in denoising diffusion probabilistic models (DDPM) by Ho et al. (2020), which demonstrated state-of-the-art performance in image generation, while drawing connections to score matching techniques (Song & Ermon (2020)). A unified theoretical framework, based on stochastic differential equations (SDEs), further integrated score matching and the denoising framework (Song et al. (2021)). Various generative models build up on this framework, improving its performance (Dhariwal & Nichol (2021a); Karras et al. (2022)).

Despite their success, diffusion models exhibit limitations, such as requiring a large number of steps (Ho et al. (2020)) and empirically struggling with imbalanced datasets (Zhang et al. (2024); Yoon et al. (2023)), often leading to mode collapse (Dhariwal & Nichol (2021a); Deasy et al. (2022)). Tackling heavy-tailed datasets also presents a challenge for diffusion models, as the finite-time diffusion processes generate finite variance data distributions, which are ill-suited for modeling heavy-tailed data. Such datasets, like financial datasets, could benefit from extensions beyond the Gaussian setting, as demonstrated in Borak et al. (2005). Moreover, although many techniques exist to improve quality at the expense of diversity (Dhariwal & Nichol (2021b); Song et al. (2023)), by only using Gaussian noise, the methods typically require highly nontrivial setups in order to obtain improved diversity while maintaining a reasonable quality (e.g., FID score) (Sadat et al. (2024); Nobis et al. (2024)).

Several approaches have been explored to address the limitations of diffusion models, particularly through the use of non-Gaussian, heavy-tailed noise distributions. The motivation behind this is

that heavy-tailed distributions, which can take on larger values, may reduce the number of sampling steps and better capture multimodal data distributions by identifying isolated modes through large noise injections (Yoon et al. (2023)). These distributions are also capable of modeling extreme events or rare occurrences in the tails, making them suitable for tasks such as audio generation (Chen et al. (2020); Kong et al. (2021)), where rare but important variations in amplitude or pitch (e.g., prosodies) can enhance sample quality and diversity. An early attempt at using heavy-tailed noise distributions was made by Nachmani et al. (2021), who replaced Gaussian noise with Gamma-distributed noise, reducing the number of diffusion steps and improving data diversity. Similarly, Deasy et al. (2022) employed generalized Gaussian distributions in a score-matching formulation to improve robustness on unbalanced datasets. However, despite these promising directions, the performance boosts are limited, since both approaches fail to provide a time-reversal formula, either using annealed Langevin dynamics when the score is available, or using heuristics based on the Gaussian approach.

Very recently, partially inspired by Simsekli (2017); Huang et al. (2020b), who employ heavy-tailed SDEs in Monte Carlo sampling for challenging distributions, Yoon et al. (2023) extended the SDE framework by replacing the light-tailed Brownian motion with a heavy-tailed driving process, introducing the Lévy-Itô model (LIM), improving performance on image data, particularly for unbalanced datasets, offering gains in metrics like FID and diversity. The tail index α controls the degree of tail heaviness, enabling tunable performance based on the characteristics of the data. When $\alpha = 2$ the noising process reduces to the Brownian motion (light-tailed), however, whenever $\alpha < 2$, the process becomes heavy-tailed with infinite variance (notably, heavier-tailed than the distributions explored by Nachmani et al. (2021), Deasy et al. (2022)).

While injecting heavy-tailed, infinite-variance noise might seem natural when the data distribution itself is also heavy-tailed, Yoon et al. (2023) further illustrated that the heavy-tailed noise can also be beneficial when sampling from compactly supported data distributions (e.g., generating images), especially in the presence of class imbalances (i.e., the large 'jumps' introduced by the heavy tails can help finding weakly represented modes). Indeed, perhaps being counter-intuitive, it has been shown that a heavy-tailed process can indeed converge to a light-tailed distribution with appropriate care Simsekli (2017); Simsekli et al. (2020); Huang et al. (2020a), making them suitable for sampling from a broad range of data distributions.

Motivation. While LIM demonstrates promising results, the technical complexity of the time-reversed SDE presents significant challenges. The Lévy process with $\alpha < 2$ has discontinuous paths and no variance, preventing the use of standard analysis tools. The proof techniques rely on fractional calculus and estimations for pseudo-differential operators, which might not be accessible for the broader community of diffusion-based generative models. While the theory is elegant, we argue that the highly technical nature of LIM, originated due to the use of continuous-time processes, might hinder its development. For instance, it is highly non-trivial to use arbitrary noise schedules.

Moreover, the loss function used in LIM presents some shortcomings: the theory requires a squared ℓ_2 loss, assuming the loss remains finite for the considered neural network. However, this may not always hold, as the noise term is heavy-tailed and admits no variance, potentially leading to infinite loss values. As a result, LIM experiments must revert to an ℓ_1 loss for stable training, suggesting that the original loss function may indeed be unworkable. Additionally, LIM is constrained to isotropic noise, further limiting its flexibility.

To overcome these issues, we propose a simpler yet effective alternative for incorporating heavy-tailed distributions, focusing on a discrete-time framework that leverages more elementary mathematical tools while maintaining performance improvements over Gaussian-based models.

Contributions. As opposed to LIM which extended the SDE-based framework, here, we take a step back and directly work on the discrete-time DDPM process and replace the Gaussian noise with α-stable noise. More precisely, we propose the following Markov process as noising process:

$$X_t = \gamma_t X_{t-1} + \sigma_t \epsilon_t^{(\alpha)}, \qquad (1)$$

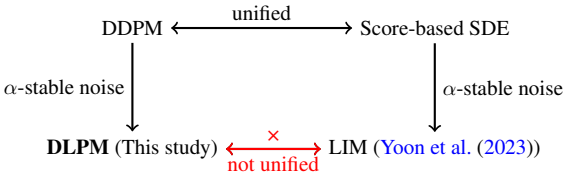


Figure 1: Illustration of available methods.

where $\epsilon_t^{(\alpha)}$ follows a multivariate α-stable distribution. When $\alpha = 2$, the process recovers the standard Gaussian DDPM, but for $\alpha < 2$, it introduces heavy-tailed noise with infinite variance. A comparison between DLPM and LIM is provided in Figure 1. Our contributions are as follows.

- **Simplified mathematical framework.** Leveraging a property of stable distributions (cf. Theorem 1), we decompose $\epsilon_t^{(\alpha)}$ into a product of a one-dimensional random variable and a Gaussian vector. This transformation reduces the forward process to a Gaussian one with modified scaling, allowing us to approximate the reverse process using only elementary tools. We call the resulting generative model Denoising Lévy Probabilistic Model (DLPM).
- **Extension to deterministic sampling.** Building on the DDIM framework (Song et al. (2020)), we introduce a deterministic sampler for DLPM, termed DLIM, which further reduces the number of sampling iterations, boosting efficiency.
- **Compatibility with existing methods.** DLPM maintains compatibility with existing DDPM implementations, requiring only minor modifications, making it a practical and flexible alternative to LIM. This simplicity extends for instance to noise schedules, which are difficult to manipulate in the continuous-time LIM framework.
- **Distinct Algorithms from LIM.** Unlike the Gaussian case, where DDPM and the score-based SDE formulation are two sides of the same coin (Song et al. (2021)), we demonstrate that DLPM and LIM result in different training algorithms and distinct backward processes. Indeed, heavy-tail distributions should prevent the simple use of the square ℓ_2 loss, as they admit no variance, problem which DLPM carefully addresses. These differences lead to unique algorithms for each method, highlighting the divergence between the two approaches.
- **Improved performances.** Thanks to our conditionally Gaussian representation strategy, our networks are only modeling conditional densities given the heavy-tailed variables. Hence the network is not learning a heavy-tailed distribution directly, but a light-tailed conditional distribution, intuitively an easier task. These fundamental differences work in favor of DLPM across several performance aspects, particularly where heavy-tailed noise injections are already known to offer advantages. Our experiments show that DLPM provides (i) better coverage of the tails of the data distribution, (ii) improved generation of unbalanced datasets, and (iii) faster computation times, as it requires significantly fewer backward steps.

2 Background on α-stable distributions

The family of α-stable distributions appears as the limiting distribution in the generalized central limit theorem (Gnedenko & Kolmogorov (1968)). In the one dimensional case, an α-stable distributed random variable X is defined through its characteristic function (Samorodnitsky et al. (1996)): for $u \in \mathbb{R}$

$$\mathbb{E}\left[\mathrm{e}^{\mathrm{i}uX}\right] = \exp\{\mathrm{i}u\mu - |\sigma u|^\alpha(1 - \mathrm{i}\varphi\beta\,\mathrm{sgn}(u))\}\ , \quad \text{where } \varphi = \begin{cases} \tan(\pi\alpha/2) & \text{if } \alpha \neq 1 \\ -(2/\pi)\log|\sigma u| & \text{otherwise} \end{cases}.$$

Here, (i) $\mu \in \mathbb{R}$ is the location parameter (ii) $\alpha \in (0, 2]$ is the tail index (iii) $\sigma > 0$ the scale parameter (iv) $\beta \in [-1, 1]$ determines the right- or left-skewness, and sgn is the sign function. We denote the α-stable distribution by $\mathcal{S}_{\alpha,\beta}(\mu, \sigma)$.

In the case where $\alpha < 1$ and $\beta = 1$, the support of the distribution becomes the positive real line (i.e., the random variable is positive), hence we call this distribution 'positive stable'. On the other hand, in the case where $\beta = 0$, the distribution $\mathcal{S}_{\alpha,0}(\mu, \sigma)$ is symmetric around μ, and denoted by $\mathcal{S}_\alpha(\mu, \sigma)$. Furthermore, in the case $\alpha = 2$, the distribution reduces to a Gaussian $\mathcal{S}_\alpha(\mu, \sigma) = \mathcal{N}(\mu, 2\sigma^2)$, hence it is light-tailed. However, whenever $\alpha < 2$, $\mathcal{S}_\alpha(\mu, \sigma)$ has heavy tails, *i.e.*, the decay rate of its tail distribution satisfies $\mathbb{P}(|X| > r) \sim r^{-\alpha}$ as $r \to \infty$ (see (Nolan, 2020, Theorem 1.2)). This implies that $\mathbb{E}[|X|^p] < \infty$ if and only if $p < \alpha < 2$.

As opposed to Gaussians, there are multiple ways of extending the α-stable distributions to the multivariate setting. In this paper, we will be interested in two major cases: (i) the isotropic (also called rotationally invariant)[1] and (ii) the non-isotropic with independent components. These distributions

[1]The noise distribution used in LIM is the isotropic α-stable distribution. Note that our framework allows for different types of α-stable distributions by following a single mathematical recipe.

are also defined through their respective characteristic functions. The random variable $X \in \mathbb{R}^d$ is isotropic α-stable if its characteristic function is given by: for all $u \in \mathbb{R}^d$, $\mathbb{E}[\exp(iu^\top X)] = \exp(i\mu^\top u - \sigma^\alpha \|u\|^\alpha)$, where $\mu \in \mathbb{R}^d$ is the location parameter and $\sigma \mathrm{I}_d$ plays the role of a covariance matrix[2]. We denote it by $X \sim \mathcal{S}^{\mathrm{i}}_\alpha(\mu, \sigma \mathrm{I}_d)$. Similarly, X follows the non-isotropic α-stable distribution $\mathcal{S}^{\mathrm{n}}_\alpha(\mu, \sigma \mathrm{I}_d)$, if for any $u \in \mathbb{R}^d$, $\mathbb{E}[\exp(iu^\top X)] = \exp(i\mu^\top u - \sigma^\alpha \sum_{i=1}^d |u_i|^\alpha)$. While both of these distributions share similar characteristics, such as having power-law tails with the same exponent, the components of the isotropic case are dependent, which results in a significant difference compared to the non-isotropic case, which has independent coordinates. When $\alpha = 2$ both options coincide with a multivariate Gaussian.

The following property of stable distributions will form the backbone of our algorithm.

Theorem 1 (See (Samorodnitsky et al., 1996, Equation 2.5.3)). *Let $\alpha < 2$, and let $X \sim \mathcal{S}^i_\alpha(\mu, \sigma \mathrm{I}_d)$. Then, $X \overset{d}{=} \mu + \sigma A^{1/2} G$, where $\overset{d}{=}$ denotes equality in distribution, $A \sim \mathcal{S}_{\alpha/2,1}(0, c_A)$ is a one-dimensional positive stable random variable with $c_A := \cos^{2/\alpha}(\pi\alpha/4)$, and $G \sim \mathcal{N}(0, \mathrm{I}_d)$.*

This theorem shows that a zero-mean, unit-scale isotropic stable random-vector can be equivalently written as the product of a *one dimensional* positive stable random variable and a standard Gaussian random vector. This fundamental property will have a significant impact in terms of incorporating α-stable noise in DDPMs in a simple way as, *conditioned on* A, the distribution of X is just a Gaussian. We conclude this section by noting that a similar decomposition for the non-isotropic case: if $X \sim \mathcal{S}^{\mathrm{n}}_\alpha(\mu, \sigma \mathrm{I}_d)$, then $X \overset{d}{=} \mu + \sigma\, \mathbf{A}^{1/2} \odot G$, where $\odot$ is the component-wise multiplication and $\mathbf{A}^{1/2} = \{A_i^{1/2}\}_{i=1}^d \in \mathbb{R}^d$ is a vector with i.i.d. components with $A_i \sim \mathcal{S}_{\alpha/2,1}(0, c_A)$.

3 Denoising Lévy Probabilistic Models

In this section, we develop our algorithm by following a similar route to the development on DDPM: we identify the forward and backward processes associated with (1) and construct a variational approximation for the backward chain for sampling. Here, we focus on isotropic α-stable noise; however, adaptation to non-isotropic α-stable noise is straightforward, as all our derivations rely on Theorem 1, hence it is omitted.

3.1 Markovian forward process

Recall that DLPM is based on the following recursion (restatement of (1)):

$$X_0 \sim p_\star \ , \quad \text{and} \quad X_t = \gamma_t X_{t-1} + \sigma_t \epsilon_t^{(\alpha)}, \tag{2}$$

where $p_\star$ is the data distribution and $\{\epsilon_t^{(\alpha)}\}_{t=1}^T$ are independent and distributed as $\mathcal{S}^{\mathrm{i}}_\alpha(0, \mathrm{I}_d)$. Thanks to the 'stability' property of α-stable distributions, *i.e.*, the sum of two α-stable random variables is still α-stable (see Appendix B for details), we can explicitly characterize the conditional distribution of X_t given X_0. Setting for any $t \in \{1, \dots, T\}$, $\gamma_{1\to t} := \prod_{i=1}^t \gamma_t$, and $\sigma_{1\to t} := (\sum_{i=1}^t (\gamma_{1\to t}\sigma_i/\gamma_{1\to i})^\alpha)^{1/\alpha}$, we show in Proposition 4 that:

$$X_t \overset{d}{=} \gamma_{1\to t} X_0 + \sigma_{1\to t} \varepsilon_t \ ,$$

where $\varepsilon_t \sim \mathcal{S}^{\mathrm{i}}_\alpha(0, \mathrm{I}_d)$. Similarly to DDPM, the noising schedule parameters $\{(\gamma_t, \sigma_t)\}_{t=1}^T$ and the horizon T are set so that the final distribution of X_T is approximately equal to $\mathcal{S}^{\mathrm{i}}_\alpha(0, \sigma_{1\to t}\mathrm{I}_d)$, choosing either (i) $\gamma_{1\to t} \to 0$ as t increases, or (ii) $\gamma_{1\to t} = 1$ and $\sigma_{1\to t}$ increasing with t. Following the terminology used in DDPM[3], we refer to schedule (i) as scale preserving and schedule (ii) as scale exploding.

3.2 Generative process

Once the forward process is run for large enough time-steps T, it is clear that X_T will be approximately stable-distributed. Hence, to go back to the data distribution $p_\star$, we now need to time-revert

[2] in this isotropic case the components are not independent even though the covariance matrix is diagonal.

[3] In the DDPM literature, these noising schedules are referred to as variance preserving, or variance exploding. We use the term 'scale' instead of the variance here, since the variance does not exist when $\alpha < 2$.

the forward process so that the reversed process can take some α-stable noise and generate a sample from $p_\star$, which is our ultimate goal. More precisely, we aim to find a backward process associated to the Markov chain $\{X_t\}_{t=0}^T$, *i.e.*, a Markov chain $\{\overleftarrow{X}_t\}_{t=0}^T$ such that the two processes $\{\overleftarrow{X}_{T-t}\}_{t=0}^T$ and $\{X_t\}_{t=0}^T$ have the same distributions. Since, by (Nolan, 2010, Theorem 1.9), any non-degenerate α-stable distribution has a smooth density with respect to the Lebesgue measure, the Markov chain (2) admits a transition density denoted by $k_{t|t-1}^{(\alpha)}$. In addition, X_t admits a density as well, denoted by $p_t^{(\alpha)}$. Then, it can be easily verified that a Markov process starting from $p_T^{(\alpha)}$ and with transition densities, for $t \in \{0, \dots, T-1\}$, $\overleftarrow{k}_{t-1|t}^{(\alpha)}(x_{t-1}|x_t) \propto p_{t-1}^{(\alpha)}(x_t) k_{t|t-1}^{(\alpha)}(x_t, x_{t-1})$ for any x_{t-1}, x_t, is a backward process associated with $\{X_t\}_{t=0}^T$, where $\propto$ denotes equality up to a normalization constant. As in the case of DDPM, this backward transition densities are unfortunately intractable, hence, we will develop a variational scheme for their approximation.

3.2.1 Approximation of the backward transition densities in DDPM

To ease the introduction of our approach, let us first recall the strategy taken in DDPM, which approximates the backward kernels by relying on a variational approximation for $\overleftarrow{k}_{0:T}^{(\alpha)}(x_{0:T}) := p_T^{(\alpha)}(x_T) \prod_{t=T}^1 \overleftarrow{k}_{t-1|t}^{(\alpha)}(x_{t-1}|x_t)$, where $\alpha = 2$ and $x_{0:T} := (x_0, \dots, x_T) \in \mathbb{R}^{d \times (T+1)}$. More precisely, the goal is to find the 'closest' distribution to $\overleftarrow{k}_{0:T}^{(\alpha)}$ in a family of distributions $\{\overleftarrow{q}_{0:T}^{\theta} : \theta \in \Theta\}$, indexed by a parameter θ taking values in some parameter space Θ (typically taken as a neural network).

The variational family is assumed to have the same decomposition as $\overleftarrow{k}_{0:T}^{(\alpha)}(x_{0:T})$, thus such that $\overleftarrow{q}_{0:T}^{\theta}(x_{0:T}) := \overleftarrow{q}_T^{\theta}(x_T) \prod_{t=T}^1 \overleftarrow{q}_{t-1|t}^{\theta}(x_{t-1}|x_t)$, where $\overleftarrow{q}_T^{\theta}$ is chosen to be the density of $\mathcal{N}(0, \sigma_{1\to t}^2 \mathrm{I}_d)$ as an approximation of $p_T^{(\alpha)}$. Then, θ is obtained by minimizing the following objective function (Ho et al., 2020, Equation 5):

$$\mathscr{L}^{\mathrm{D}}(\theta) := \sum_{t=2}^T \mathscr{L}_{t-1}^{\mathrm{D}}(\theta) \quad \text{with} \quad \mathscr{L}_{t-1}^{\mathrm{D}}(\theta) = \mathbb{E}[\mathrm{KL}(k_{t-1|0,t}^{(\alpha)}(\cdot|X_0, X_t) \| \overleftarrow{q}_{t-1|t}^{\theta}(\cdot|X_t))] \ , \tag{3}$$

where KL denotes the Kullback-Leibler divergence and $k_{t-1|0,t}^{(\alpha)}$ denotes the conditional density of X_{t-1} given X_0 and X_t. As $\alpha = 2$ in this case, $k_{t-1|0,t}^{(\alpha)}$ is Gaussian (Ho et al., 2020, Equation 6,7), motivating the choice of Gaussian densities $\overleftarrow{q}_{t-1|t}^{\theta}(x_{t-1}|x_t)$ as elements of the variational family at hand, since one obtains a closed-form formula for the KL terms, *i.e.*,

$$\overleftarrow{q}_{t-1|t}^{\theta}(x_{t-1}|x_t) = \phi_d\left(x_{t-1}|\hat{\mathtt{m}}_{t-1}^{\theta}(x_t), \hat{\Sigma}_{t-1}^{\theta}(x_t)\right) \ , \tag{4}$$

where $(x, \mathtt{m}, \Sigma) \mapsto \phi_d(x|\mathtt{m}, \Sigma)$ is the density of the d-dimensional Gaussian distribution with mean $\mathtt{m}$ and covariance matrix Σ, and $\hat{\mathtt{m}}_{t-1}^{\theta}, \hat{\Sigma}_{t-1}^{\theta}$ are functions of x_t parameterized by θ. This approach relies on the fact that $k_{t-1|0,t}^{(\alpha)}$ is analytically tractable. Unfortunately, when $\alpha < 2$, it is not the case anymore. We now expose our methodology to address this limitation.

3.2.2 A data augmentation approach

To obtain a tractable objective function for learning a variational approximation of the backward transition densities, we rely on a data augmentation approach, which is a classical MCMC technique (see Brooks et al. (2011), Chapter 10). Consider the Markov chain $Y_0 \sim p_\star$, and for $t \in \{1, \dots, T\}$,

$$Y_t = \gamma_t Y_{t-1} + \sigma_t A_t^{1/2} G_t \ , \tag{5}$$

where $\{G_t\}_{t=1}^T$ and $\{A_t\}_{t=1}^T$ are independent random variables, distributed according to $G_t \sim \mathcal{N}(0, \mathrm{I}_d)$, and $A_t \sim \mathcal{S}_{\alpha/2,1}(0, c_A)$ with $c_A = \cos^{2/\alpha}(\pi\alpha/4)$. From Theorem 1, $\{Y_t\}_{t=0}^T$ is a Markov chain that admits the same distribution as $\{X_t\}_{t=0}^T$. As a result, *conditioned on* $\{A_t\}_{t=1}^T$ and Y_0, $\{Y_t\}_{t=1}^T$ is a Markov chain with Gaussian transition densities:

$$k_{1:T|0,a}^{(\alpha)}(y_{1:T}|y_0, a_{1:T}) = \prod_{t=1}^T \phi_d(y_t|\gamma_t y_{t-1}, \sigma_t^2 a_t) \ .$$

Again, we can explicitly characterize the conditional distribution of Y_t given $Y_0, \{A_t\}_{t=1}^T$. Setting for any $t \in \{1, \dots, T\}$, $\Sigma_{1\to t}(a_{1:t}) := \sum_{k=1}^t (\gamma_{1\to t} a_k^{1/2} \sigma_k / \gamma_{1\to k})^2$, we show in Proposition 5 that:

$$Y_t \overset{d}{=} \gamma_{1\to t} Y_0 + \Sigma_{1\to t}(A_{1:t}) \varepsilon_t \ , \tag{6}$$

where $\varepsilon_t \sim \mathcal{S}_\alpha^{\mathrm{i}}(0, \mathrm{I}_d)$. We propose approximating the backward process associated to $\{Y_t\}_{t=0}^T$, *given* $\{A_t\}_{t=1}^T$, adapting the DDPM approach. This time, for the backward process, we use the conditional density of Y_{t-1} given Y_0, Y_t and $A_{1:T}$:

$$\overleftarrow{k}^{(\alpha)}_{1:T|0,a}(y_{1:T}|y_0, a_{1:T}) := p_T^{(\alpha)}(y_T) \prod_{t=T}^{2} k^{(\alpha)}_{t-1|0,t,a}(y_{t-1}|y_t, y_0, a_{1:t}),$$

where $k^{(\alpha)}_{t-1|0,t,a}(\cdot|y_t, y_0, a_{1:t})$ is now the tractable density of a Gaussian distribution:

Proposition 1. *The density of the backward process associated to $\{Y_t\}_{t=0}^T$ given $Y_0, \{A_t\}_{t=1}^T$, denoted by $k^{(\alpha)}_{t-1|t,0,a}(\cdot|y_t, y_0, a_{1:t})$, is the density of a Gaussian distribution $\mathcal{N}(\tilde{\mathtt{m}}_{t-1}, \tilde{\Sigma}_{t-1})$, with mean $\tilde{\mathtt{m}}_{t-1}$ and variance $\tilde{\Sigma}_{t-1}$ equal to:*

$$\tilde{\mathtt{m}}_{t-1}(y_t, y_0, a_{1:t}) = \frac{1}{\gamma_t}\left(y_t - \Gamma_t(a_{1:t})\sigma_{1\to t}\epsilon_t(y_t, y_0)\right) \ , \ \tilde{\Sigma}_{t-1}(a_{1:t}) = \Gamma_t(a_{1:t})\Sigma_{1\to t-1}(a_{1:t-1}) \ ,$$

where $\Sigma_{1\to t}$ is as in (6), $\Gamma_t = 1 - \gamma_t \Sigma_{1\to t-1}/\Sigma_{1\to t}$, and $\epsilon_t(y_t, y_0) = (y_t - \gamma_{1\to t} y_0)/\sigma_{1\to t}$.

See Appendix C.3 for the derivations required for Proposition 1. For our generative model, we reconsider the family of Gaussian variational approximation introduced in (4), modified to account for an iid. sequence $\{A_t\}_{t=1}^T$: $\overleftarrow{q}^{\theta}_{0:T}(y_{0:T}) := \int \overleftarrow{q}^{\theta}_T(y_T) \prod_{t=T}^1 \overleftarrow{q}^{\theta}_{t-1|t,a}(x_{t-1}|x_t, a_{1:t}) \psi^{(\alpha)}_{1:T}(a_{1:T}) \mathrm{d}a_{1:T}$, where $\psi^{(\alpha)}_{1:T}(a_{1:T}) = \prod_{t=1}^T \psi^{(\alpha)}(a_t)$ and $\psi^{(\alpha)}$ denotes the density of $\mathcal{S}_{\alpha/2,1}(0, c_A)$, and

$$\overleftarrow{q}^{\theta}_{t-1|t,a}(y_{t-1}|y_t, a_{1:t}) = \phi_d(y_{t-1}|\hat{\mathtt{m}}^{\theta}_{t-1}(y_t, a_{1:t}), \hat{\Sigma}^{\theta}_{t-1}(y_t, a_{1:t})) \ .$$

3.3 Variational Inference Objective

We consider the following loss function:

$$\mathscr{L}^{\mathrm{L}}(\theta) := \mathbb{E}\left[\sum_{t=2}^T \left(\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:T})\right)^r\right], \qquad \text{where} \tag{7}$$

$$\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t}) := \mathbb{E}\left[\mathrm{KL}\left(k^{(\alpha)}_{t-1|t,0,a}(\cdot|Y_t, Y_0, A_{1:t}) \parallel \overleftarrow{q}^{\theta}_{t-1|t,a}(\cdot|Y_t, A_{1:t})\right) \middle| A_{1:t}\right] \ ,$$

and $r > 0$, $k^{(\alpha)}_{t-1|0,t,A}$ denotes the conditional density of Y_{t-1} given Y_0, Y_t and $A_{1:T}$. In order to ensure that the expectations with respect to $A_{1:T}$ are finite, we need to choose $r < \frac{\alpha}{2}$ when $\alpha \in (1, 2)$. For simplicity, in the rest of the paper, we will use $r = \frac{1}{2}$.

A comparison with DDPM's loss function $\mathscr{L}^{\mathrm{D}}$ (cf. (3)) immediately illustrates that, thanks to Theorem 1, $\mathscr{L}^{\mathrm{L}}$ is almost identical to $\mathscr{L}^{\mathrm{D}}$ up to taking expectations with respect to one-dimensional random variables and the taking square-root of the summands[4]. We show in Appendix C.4.3 that, alike DDPM, our loss is obtained from a KL minimization principle, serving as an upper bound to:

$$\mathbb{E}\left[\left(\mathrm{KL}(k^{(\alpha)}_{0|a}(\cdot|A_{1:T}) | \overleftarrow{q}^{\theta}_{0|a}(\cdot|A_{1:T}))\right)^r\right] \ ,$$

when $r \in (0, 1]$. This additionally shows how any zero of the loss corresponds to a perfect generative model, while maintaining a similar objective function. The crucial property of (7) is that, since both $k^{(\alpha)}_{t-1|t,0,a}$ and $\overleftarrow{q}^{\theta}_{t-1|t,a}$ are Gaussian (thanks to the conditioning), the KL term admits a closed-form analytical formula, as in the case of DDPM.

From a practical perspective, (7) suggests that we can use the same software architecture as for DDPM, with a slight modification to compute the outer expectation, which can be simply estimated by a Monte Carlo, or median-of-means procedure (Lugosi & Mendelson (2019)). In order to obtain

[4] We note that, with the choices of $\alpha = 2$ and $r = 1$, we exactly recover DDPM.

a final denoising training loss, we provide three design choices, for which the full details are given in Appendix C.4. They are similar to what is classically done in diffusion models (Ho et al. (2020); Nichol & Dhariwal (2021); Karras et al. (2022)):

D1. We set a fixed variance $\hat{\Sigma}_t^\theta = \tilde{\Sigma}_t$ for the reverse process.

D2. We reparameterize the model to predict the value of $\epsilon_t(y_t, y_0)$ with a network $\hat{\epsilon}_t^\theta$, setting

$$\hat{\mathtt{m}}_{t-1}^\theta(Y_t, A_{1:t}) = \frac{1}{\gamma_t}\left(Y_t - \sigma_{1\to t}\Gamma_t(A_{1:t})\hat{\epsilon}_t^\theta(Y_t, A_{1:t})\right) .$$

Moreover, we drop the dependency of $\hat{\epsilon}_t^\theta$ on $\{A_t\}_{t=1}^T$, making $\hat{\epsilon}_t^\theta$ only a function of Y_t. This enables re-using classical diffusion models network architectures.

D3. Assuming D1, D2, we obtain $\mathscr{L}_{t-1}^{\mathrm{L}}(\theta) = \mathbb{E}\left[\lambda_{t,A_{1:t}}^2 \|\hat{\epsilon}_t^\theta(Y_t, A_{1:t}) - \epsilon_t(Y_t, Y_0)\|^2\right]$ where $\lambda_{t,a_{1:t}} = \Gamma_t(a_{1:t})\sigma_{1\to t}/2\gamma_t\tilde{\Sigma}_t$, and $\epsilon_t(y_t, y_0) = (y_t - \gamma_{1\to t}y_0)/\sigma_{1\to t}$. We then fix $\lambda_{t,a_{1:t}} = 1$.

Proposition 2 (Simplified denoising loss)**.** *With the design choices D1, D2, D3, we obtain the simplified denoising objective function:*

$$\mathscr{L}^{\mathrm{Simple}}(\theta) = \sum_{t=1}^{T} \mathbb{E}\left[\mathbb{E}\left(\|\hat{\epsilon}_t^\theta(Y_t) - \epsilon_t(Y_t, Y_0)\|^2 \mid A_{1:t}\right)^{1/2}\right] , \tag{8}$$

where the model $\hat{\epsilon}_t^\theta$ is designed to fit the noise $\epsilon_t(Y_t, Y_0) = (Y_t - \gamma_{1\to t}Y_0)/\sigma_{1\to t}$ added at time-step t, and $\gamma_{1\to t}, \sigma_{1\to t}$ are as given in Proposition 4. Thus the model is not learning a heavy-tailed distribution directly, but the light-tailed conditional distribution.

See Appendix C.4.2 for the derivations required for Proposition 2. Finally, in Appendix C.5, we show that on some conditions, satisfied under design choices D1, D2, D3, the expectation of each term in $\mathscr{L}^{\mathrm{L}}$ can be rewritten as an expectation with respect to only one univariate random variable (as opposed to t variables, i.e., $A_{1:t}$), reducing the additional computational burden of accommodating heavy tails. As we will estimate the expectations by Monte Carlo averaging, reducing the number of random variables in the expectation is equally important to reduce the error in the estimation. The resulting loss is given in Proposition 9.

3.4 Deterministic sampling

Using the same techniques as in denoising diffusion implicit models (DDIM, Song et al. (2020)), we can recover a deterministic sampling scheme. We call this algorithmic extension Denoising Lévy Implicit Models (DLIM), which details are given in Appendix D. We recapitulate our various sampling and training algorithms in Appendix A. We note that by using an ordinary differential equation (ODE) approach, LIM was extended similarly by Yoon et al. (2023), obtaining LIM-ODE, which encompasses a deterministic sampling scheme.

3.5 Comparing DLPM to LIM

The objective function (7) is slightly different from the one obtained in the continuous setting of LIM by Yoon et al. (2023). The training equations are very similar, and can be reformulated to involve a denoising loss (see Appendix E.1, and (28)):

$$\mathcal{L}_{t-1} : \theta \mapsto \mathbb{E}\left(\|\hat{\epsilon}_t^\theta(X_t) - \epsilon_t(X_t, X_0)\|_p^\eta\right) .$$

As a refresher, in the case of DDPM, one sets $p = 2, \eta = 2$. In the case of DLPM, our discussion leads us to the choice $p = 2, \eta = 1$ (see (8)). In the case of LIM, the theory relies on a squared ℓ_2 loss, setting $p = 2, \eta = 2$ in order to properly derive the loss and effectively approximate the true score of the data, at various noisescales. One must therefore make the assumption that $\mathcal{L}_{t-1}$ is not infinite for each parameter θ considered, which may not hold since $\epsilon_t(X_t, X_0)$ is α-stable distributed and admits no variance. In the experiments for LIM, one is forced to revert to an ℓ_1 loss, by setting $\eta = 1, p = 1$, to obtain a stable training, potentially indicating that the loss is infinite.

Additionally, DLPM and LIM yield different backward processes, which in turn lead to distinct sampling algorithms – cf. Table 4 in the Appendix. Finally, the LIM framework can only accommodate isotropic noise. We refer to Appendix E.1 for a detailed comparison between DLPM and LIM.

4 EXPERIMENTS

After the groundwork of the previous sections, we design experiments to demonstrate the practical strengths of our DLPM approach as compared to LIM, apart from its technical simplicity. We recall that setting $\alpha = 2$ simply reverts LIM to classical diffusion, and DLPM reverts to DDPM, appart from the square-root in the loss function. As specified in (Yoon et al. (2023), Appendix G), LIM relies on gradient and noise clipping, which introduces extra hyperparameters that must be fine-tuned for each dataset. In the experiments, we use these clipping parameters only when specified. The experimental details relative to this section are available in Appendix G.

As our loss function (7) involves an expectation with respect to $A_{1:T}$, we propose estimating it by using the *median-of-means* estimator, which is known to have better performance for heavy-tailed distributions (Lugosi & Mendelson (2019)). For an integer M, this approach requires sampling M^2 many $A_{1:T}$ terms, then split them into M groups of size M. To approximate the expectation, we take the sample mean of each group, and finally take the median of the computed M sample means. In our experiments, we explore $M = 1$ (approximating the expectation with only one sample), denoted simply by DLPM, and $M = 5$, denoted by DLPM$_5$. Similarly, we denote DLIM and DLIM$_5$ for the corresponding deterministic sampling schemes.

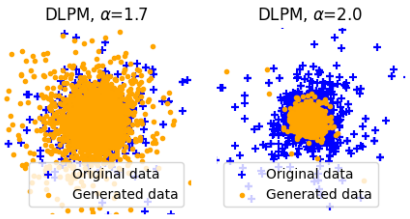
Finally, we consider the range $1.5 \leqslant \alpha \leqslant 2.0$. In our experiments on images, we make use of the dataset CIFAR10_LT (long tail), that has been introduced in Yoon et al. (2023) as an unbalanced modification of the CIFAR10 dataset.

4.1 DATA COVERAGE AND MODE COLLAPSE IN TWO-DIMENSIONAL DATA

Before progressing to higher dimensional problems, we start with easily controlled and visualized two-dimensional datasets, in order to validate the competitiveness of our method in the contexts where heavy-tailed diffusions are of interest. In particular, we consider heavy-tailed and unbalanced multi-modal datasets. See Appendix G.1 for details about the experimental setup in these settings.

Enhancing data coverage: capturing the tail of the distribution. We set $d = 2$ and generate data points distributed as $\mathcal{S}^{\mathrm{i}}_{\alpha}(0, 0.05 \cdot \mathrm{I}_d)$, with $\alpha = 1.7$. Our aim is to test the ability of each method to cover the dataset correctly; the main challenge is to correctly capture the tails. As we can visually observe in Figure 2, the backward diffusion process in the Gaussian case cannot produce truly heavy-tailed data, mainly stemming from the fact that its variance is always finite. As expected, we see improvements when using noise with $\alpha < 2$.

Figure 2: DLPM with $\alpha = 1.7$ and $\alpha = 2.$. The lighter-tailed process fails to capture the distribution's tail.



To quantify this behaviour, we utilize the mean square logarithmic error (MSLE) metric, which compares the relative error between the higher quantiles (*i.e.*, the quantiles corresponding to the tail region that has probability $1-\xi$) of the true and the generated data distribution (see Appendix G.3 for detailed definitions). We observe in Table 1 how, as α gets smaller, one gets better tail approximation. Furthermore, DLPM consistently outperforms LIM, indicating that the generation process benefits from the heavy-tailed denoising formulation, rather than the continuous-time one, in this setting.

Method	$\alpha = 1.5$	$\alpha = 1.6$	$\alpha = 1.7$	$\alpha = 1.8$	$\alpha = 1.9$	$\alpha = 2.0$
DLPM	**0.160** ± 0.128	**0.081** ± 0.078	**0.071** ± 0.028	**0.099** ± 0.044	**0.132** ± 0.101	0.798 ± 0.601
DDPM	-	-	-	-	-	0.528 ± 0.400
						1.0e-1
LIM	0.743 ± 0.290	0.497 ± 0.311	0.267 ± 0.077	0.653 ± 0.413	2.444 ± 1.067	1.239 ± 0.240
	1.0e-08	*8.6e-06*	*1.3e-10*	*8.8e-06*	*7.9e-09*	*5.0e-3*

Table 1: MSLE$_{\xi=0.95}$ $\downarrow$ averaged over 20 runs. Figures below scores corresponds to p-values from Welch's t-test (assuming unequal variances), comparing the mean of DLPM with the given method.

Enhancing data coverage: addressing mode collapse. To assess the robustness of DLPM to mode collapse, we consider an unbalanced mixture of nine Gaussian distributions. We set their standard deviation to 0.05 and arrange them in a grid-like pattern with equal spacing, in the square $[-1, 1]^2$. Their mixture weights range from .01 to .3 [5]. We use the F_1^{pr} score (*i.e.*, the harmonic mean of precision and recall, see Appendix G.3) to assess, in a single summary statistic, the quality and diversity in the generated data. As shown in Table 2, we are able to achieve improved scores by choosing a tail index $\alpha < 2$ with DLPM. This is not necessarily the case for LIM, which is consistently outperformed. Finally, DLPM_5 shows its strengths with better performance over all the range of α, though at the cost of 5 times the run time.

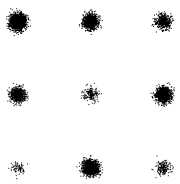


Figure 3: Gaussian grid

Method	$\alpha = 1.5$	$\alpha = 1.6$	$\alpha = 1.7$	$\alpha = 1.8$	$\alpha = 1.9$	$\alpha = 2.0$
DLPM	0.933 ± 0.018	0.923 ± 0.005	0.933 ± 0.028	0.923 ± 0.024	0.907 ± 0.034	0.862 ± 0.028
DLPM_5	**0.944 ± 0.013**	**0.943 ± 0.021**	**0.943 ± 0.010**	**0.941 ± 0.014**	**0.928 ± 0.016**	-
	9.0e-3	*1.6e-05*	*7.4e-2*	*9.0e-4*	*3.9e-3*	
LIM	0.842 ± 0.039	0.850 ± 0.046	0.868 ± 0.034	0.874 ± 0.030	0.884 ± 0.017	0.874 ± 0.027
	1.7e-14	*1.3e-09*	*5.7e-11*	*3.9e-09*	*1.9e-3*	*9.6e-2*
DDPM	-	-	-	-	-	0.867 ± 0.029
						5.0e-1

Table 2: $F_1^{\text{pr}} \uparrow$ score, averaged over 30 runs. Figures below scores corresponds to p-values from Welch's t-test (assuming unequal variances), comparing the mean of DLPM with the given method.

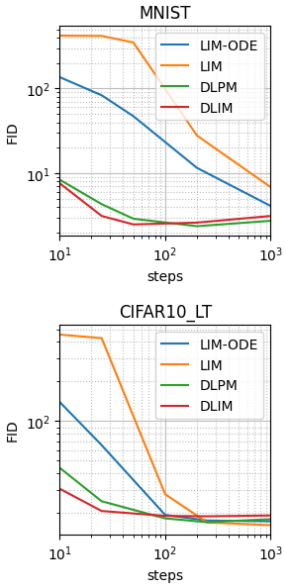
4.2 Experiments on image data

To fairly illustrate the differences between LIM and DLPM, we use the same improved DDPM neural network architecture, as designed in Nichol & Dhariwal (2021). The specific configuration for each dataset is carefully described in Appendix G.2. Our experiments are designed to compare deterministic and stochastic generation methods under varying conditions. As a visual check, examples of generated images are listed in Appendix G.4.

Table 3: FID↓, 1000 sampling steps for LIM and DLPM, 25 sampling steps for LIM-ODE and DLIM.

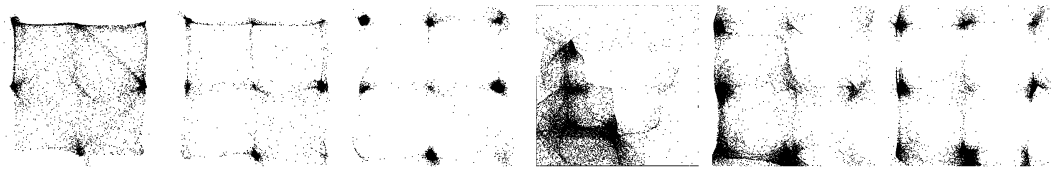
MNIST	$\alpha = 1.5$	$\alpha = 1.7$	$\alpha = 1.8$	$\alpha = 1.9$	$\alpha = 2.0$
DDPM	-	-	-	-	**3.43**
LIM	14.37	11.54	11.18	13.75	11.69
w/ clipping	*4.08*	*5.17*	*6.81*	*11.20*	
DLPM_5	**3.80**	3.03	**2.51**	**2.71**	-
DLPM	5.39	**2.94**	2.93	3.24	3.63
DDIM	-	-	-	-	**5.16**
LIM-ODE	49.63	78.59	92.93	109.48	29.04
w/ clipping	*45.72*	*68.15*	*85.09*	*113.20*	
DLIM_5	**3.37**	2.93	3.44	4.31	-
DLIM	3.38	**2.81**	**3.18**	**3.27**	5.18
CIFAR10_LT					
DDPM	-	-	-	-	**19.05**
LIM	75.38	35.15	31.14	21.68	21.56
w/ clipping	*16.13*	*16.21*	*17.67*	*19.24*	
DLPM	**16.10**	**18.00**	**19.94**	**20.21**	21.07
DDIM	-	-	-	-	**23.44**
LIM-ODE	42.07	91.64	105.95	407.79	32.00
w/ clipping	*30.17*	*65.78*	*84.55*	*101.70*	
DLIM	**20.69**	**20.77**	**21.96**	**22.79**	23.99

Figure 4: FID↓, $\alpha = 1.7$



Convergence speed. Consistent with existing literature Song et al. (2020), our findings as shown in Figure 4 confirm that deterministic generation outperforms its stochastic counterpart significantly,

[5] The exact mixture weights are $\{.01, .02, .02, .05, .05, .1, .1, .15, .2, .3\}$.



DLIM, 5 steps 10 steps 25 steps **LIM**, 5 steps 10 steps 25 steps

Figure 5: DLIM and LIM-ODE with small number of steps, on the Gaussian grid of Figure 3.

especially when fewer than 100 diffusion steps are used, on both MNIST and CIFAR10_LT. As the number of diffusion steps increases, both of these sampling methods produce similar results. This observation highlights that the advantages of the diffusion process do not only stem from increased randomness at sampling time. These heavy-tailed processes may define more appropriate vector field on which the noise is transported back to the original data distribution, which would lead to improved model performance (see Karras et al. (2022) for similar discussions on DDPM vs DDIM).

The previous observations are quantitatively supported in Table 3, where we present results for both deterministic and stochastic sampling strategies. We compare both methods on stochastic generation at a high step count, to compare their performance at their best regime, and on deterministic generation at a small step count, to assess the tradeoff in computations/quality offered by both methods. As we can see, DLPM surpasses LIM on both datasets. Moreover, these results show that LIM's performance deteriorates significantly when clipping is not used, raising questions about whether the framework of LIM is inherently well-suited for heavy-tailed distributions. More interestingly, we observe in Table 3 that DLPM consistently outperforms LIM and offers satisfying image quality at low number of steps, both for stochastic and deterministic sampling.

Generated images after 25 steps achieve a FID score of 2.81 on MNIST and of 20.69 on CIFAR10_LT, as compared to respectively 45.72 and 30.17 for LIM-ODE with clipping. On MNIST, with $\alpha = 1.7$, DLIM is able to match the sample quality of DLPM with 40 times less diffusion steps, further proving its efficacy. To visualize these behaviours, we display on Figure 5 different generation with varying time horizon T. We can see how the backward process defined by DLIM is able to approach the true data distribution more accurately.

Eventhough lower α usually entails lower FID in this table, LIM-ODE shows worse performance than the Gaussian diffusion at 25 reverse steps; since the image quality is still monotonically decreasing with α except for $\alpha = 2$, we can conjecture that the initial instability introduced by heavy-tails are finally counterbalanced by the various benefits of heavy-tailed diffusion. DLPM_5 shows consistent improvement over baseline, more particularly for stochastic sampling. We provide results for non-isotropic generation, and additional F_1^{pr} metrics on image data in Appendix G.4, further supporting our claims.

5 Conclusion

In this study, we proposed DLPM and DLIM, as heavy-tailed generalizations of DDPM and DDIM. Compared to state-of-the-art, we believe that our approach would be more accessible to the community, thanks to its elementary tools. The various experiments we have conducted suggest that DLPM is more effective in leveraging the characteristics of heavy-tailed distributions, providing robust performance across heavy-tailed data, unbalanced datasets and requiring a lower number of diffusion steps.

REFERENCES

Michaël Allouche, Stéphane Girard, and Emmanuel Gobet. EV-GAN: Simulation of extreme events with ReLU neural networks. *Journal of Machine Learning Research*, 23(150):1–39, 2022. URL https://hal.science/hal-03250663.

Szymon Borak, Wolfgang Karl Härdle, and Rafał Weron. Stable distributions. *Statistical Tools for Finance and Insurance*, 04 2005. doi: 10.1007/3-540-27395-6_1.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation, 2020. URL https://arxiv.org/abs/2009.00713.

Jacob Deasy, Nikola Simidjievski, and Pietro Liò. Heavy-tailed denoising score matching, 2022.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021a.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021b. URL https://arxiv.org/abs/2105.05233.

B. V. Gnedenko and A. N. Kolmogorov. *Limit distributions for sums of independent random variables*. Translated from the Russian, annotated, and revised by K. L. Chung. With appendices by J. L. Doob and P. L. Hsu. Revised edition. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills., Ont., 1968.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

William Holt and Duy Nguyen. Essential aspects of bayesian data imputation. *SSRN*, 2023, jun 2023. doi: 10.2139/ssrn.4494314. URL http://dx.doi.org/10.2139/ssrn.4494314.

Lu-Jing Huang, Mateusz B. Majka, and Jian Wang. Approximation of heavy-tailed distributions via stable-driven sdes, 2020a. URL https://arxiv.org/abs/2007.02212.

Lu-Jing Huang, Mateusz B. Majka, and Jian Wang. Approximation of heavy-tailed distributions via stable-driven sdes, 2020b. URL https://arxiv.org/abs/2007.02212.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis, 2021. URL https://arxiv.org/abs/2009.09761.

Gábor Lugosi and Shahar Mendelson. Mean estimation and regression under heavy-tailed distributions: A survey. *Foundations of Computational Mathematics*, 19(5):1145–1190, 2019.

Eliya Nachmani, Robin San Roman, and Lior Wolf. Denoising diffusion gamma models, 2021.

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021.

Gabriel Nobis, Maximilian Springenberg, Marco Aversa, Michael Detzel, Rembert Daems, Roderick Murray-Smith, Shinichi Nakajima, Sebastian Lapuschkin, Stefano Ermon, Tolga Birdal, Manfred Opper, Christoph Knochenhauer, Luis Oala, and Wojciech Samek. Generative fractional diffusion models, 2024. URL https://arxiv.org/abs/2310.17638.

J. P. Nolan. *Stable Distributions - Models for Heavy Tailed Data*. Birkhäuser, Boston, 2010. In progress, Chapter 1 online at academic2.american.edu/∼jpnolan.

John P Nolan. Univariate stable distributions. *Springer Series in Operations Research and Financial Engineering*, 10:978–3, 2020.

Manuel D Ortigueira, Taous-Meriem Laleg-Kirati, and J A Tenreiro Machado. Riesz potential versus fractional laplacian. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(9): P09032, sep 2014. doi: 10.1088/1742-5468/2014/09/P09032. URL https://dx.doi.org/10.1088/1742-5468/2014/09/P09032.

Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M. Weber. Cads: Unleashing the diversity of diffusion models through condition-annealed sampling, 2024. URL https://arxiv.org/abs/2310.17347.

Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall, 2018.

Gennady Samorodnitsky, Murad S Taqqu, and RW Linde. Stable non-gaussian random processes: stochastic models with infinite variance. *Bulletin of the London Mathematical Society*, 28(134): 554–555, 1996.

Umut Simsekli. Fractional Langevin Monte carlo: Exploring Levy driven stochastic differential equations for Markov chain Monte Carlo. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3200–3209. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/simsekli17a.html.

Umut Simsekli, Lingjiong Zhu, Yee Whye Teh, and Mert Gurbuzbalaban. Fractional underdamped langevin dynamics: Retargeting sgd with momentum under heavy-tailed gradient noise. In *International conference on machine learning*, pp. 8970–8980. PMLR, 2020.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *CoRR*, abs/2010.02502, 2020. URL https://arxiv.org/abs/2010.02502.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models, 2023. URL https://arxiv.org/abs/2303.01469.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Comput.*, 23(7):1661–1674, jul 2011. ISSN 0899-7667. doi: 10.1162/NECO_a_00142. URL https://doi.org/10.1162/NECO_a_00142.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications, 2024.

Eun Bi Yoon, Keehun Park, Sungwoong Kim, and Sungbin Lim. Score-based generative models with lévy processes. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 40694–40707. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8011b23e1dc3f57e1b6211ccad498919-Paper-Conference.pdf.

Tianjiao Zhang, Huangjie Zheng, Jiangchao Yao, Xiangfeng Wang, Mingyuan Zhou, Ya Zhang, and Yanfeng Wang. Long-tailed diffusion models with oriented calibration. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NW2s5XXwXU.

APPENDIX

The Appendix is organized as follows:

- In Appendix A, we provide the pseudo-code for the training and sampling algorithms of DLPM and DLIM.
- In Appendix B, we characterize the stability property of the α-stable distribution, and we give explicit formulas for the distribution of the sum of two α-stable random variables.
- In Appendix C, we provide the detailed theory and derivations of the DLPM framework. Working on the process $\{X_t\}_{t=0}^T$ defined in (2), and its data augmentation counterpart $\{Y_t\}_{t=0}^T$ defined in (5), we start in Appendix C.2 by characterizing the distribution of X_t given X_0 from a given noise schedule $\{(\gamma_t, \sigma_t)\}_{t=1}^T$, as this defines the characteristic location $\gamma_{1\to t}$ and scale $\sigma_{1\to t}$ of the process at time t. Likewise we characterize the distribution of Y_t given $Y_0, \{A_t\}_{t=1}^T$, as this defines the characteristic location $\gamma_{1\to t}$ and variance $\Sigma_{1\to t}$ of the augmented process at time t.

  In Appendix C.3, we focus on the Gaussian trick exploited by the process $\{Y_t\}_{t=0}^T$ defined in (5). This leads us to the an explicit formula for the Gaussian density of the backward process conditioned on a sequence $\{A_t\}_{t=1}^T$ of α-stable random variables.

  In Appendix C.4, we further put this characterization into good use by obtaining a closed-form formula for the loss function . Follows a discussion on design choices for the model and the loss function at hand, which leads us to the simplified training loss (8) corresponding to the denoising loss for α-stable diffusion.

  In Appendix C.4.3 we provide a more principled approach to derive the loss function at hand.

  Finally, in Appendix C.5, we provide a faster sampling strategy for training DLPM, computing each loss term $\mathscr{L}_{t-1}^{\mathrm{L}}$ using only two heavy-tailed random variables per datapoint, instead of t random variables.
- In Appendix D, we adapt the setting for deterministic sampling in classical denoising diffusion DDIM (Song et al. (2020)) to our α-stable framework. We naturally call this extension DLIM. Alike DDIM, it is true that the same neural networks can be used for both the DLIM and DLPM generation procedures.
- In Appendix E, we compare in more details the two discrete and continuous frameworks DLPM and LIM, underlining how they offer two distinct loss functions, training and sampling algorithms.
- In Appendix F, we give proofs relative to our technique for faster training, as introduced in Appendix C.5.
- In Appendix G, we provide additional experimental details.

A ALGORITHMS FOR DLPM AND DLIM

In this section, we explicitly provide the algorithms needed to train and sample from the DLPM and DLIM generative methods.

A.1 TRAINING

The same models can be shared between DLPM and DLIM, as underlined in Appendix D.3. We introduce the values $\sigma_{1\to t}, \gamma_{1\to t}$ determined by the noise schedule, as presented in Proposition 4:

$$\gamma_{1\to t} = \prod_{i=1}^{t} \gamma_t \ , \qquad \sigma_{1\to t} = \left[\sum_{i=1}^{t} \left(\frac{\gamma_{1\to t}}{\gamma_{1\to i}} \sigma_i \right)^{\alpha} \right]^{1/\alpha} .$$

We define $c_A = \cos^{2/\alpha}(\pi\alpha/4)$ as in Theorem 1. We make the design choices D1, D2, D3 for our model, as described in Appendix C.4.2. Finally, we use the method for faster sampling as described in Appendix C.5, Proposition 9. The resulting training algorithm is given in Algorithm 2.

Algorithm 2 DLPM training - simplified loss

Require: model $\{\hat{\epsilon}_t^\theta\}_{t=1}^T$, noise schedule $\{(\gamma_t, \sigma_t)\}_{t=1}^T$, data Y_0

1: Sample $t \sim \mathcal{U}[1, T]$
2: Sample $\bar{A}_t \sim \mathcal{S}_{\alpha/2,1}(0, c_A)$
3: Sample $G_t \sim \mathcal{N}(0, \mathrm{I}_d)$
4: $Y_t \leftarrow \gamma_{1\to t} Y_0 + \sigma_{1\to t} \bar{A}_t^{1/2} G_t$
5: $L_{t-1} \leftarrow \|\hat{\epsilon}_t^\theta(Y_t) - \bar{A}_t^{1/2} G_t\|_2$
return L_{t-1}

A.2 Sampling

Given the noise schedule and a sequence $\{A_t\}_{t=1}^T$ of α-stable random variables we define:

$$\Sigma_{1\to t}(A_{1:t}) = \sum_{k=1}^{t} \left(\frac{\gamma_{1\to t}}{\gamma_{1\to k}} \sqrt{A_k} \sigma_k \right)^2 , \quad \Gamma_t(A_{1:t}) = 1 - \frac{\gamma_t^2 \Sigma_{1\to t-1}(A_{1:t-1})}{\Sigma_{1\to t}(A_{1:t})} , \tag{9}$$

see Proposition 6 and Equation (12) from Proposition 5 for precise statements. We give our sampling algorithm for DLPM in Algorithm 3.

Algorithm 3 Stochastic sampling (DLPM)

Require: model $\{\hat{\epsilon}_t^\theta\}_{t=1}^T$, noise schedule $\{(\gamma_t, \sigma_t)\}_{t=1}^T$

1: Sample $Y_T \sim \mathcal{S}(0, \sigma_{1\to T} \mathrm{I}_d)$
2: Sample $\{A_t\}_{t=1}^T$ i.i.d., $A_t \sim \mathcal{S}_{\alpha/2,1}(0, c_A)$
3: **for** $t \leftarrow T$ to 1 **do**
4: Compute $\Sigma_{1\to t}, \Sigma_{1\to t-1}, \Gamma_t$ as in (9)
5: Sample $G_t \sim \mathcal{N}(0, \mathrm{I}_d)$
6: $\hat{\Sigma}_{t-1}^\theta \leftarrow \Gamma_t \Sigma_{1\to t-1}$
7: $Y_{t-1} \leftarrow \dfrac{Y_t}{\gamma_t} - \Gamma_t \sigma_{1\to t} \hat{\epsilon}_t^\theta(Y_t) + \sqrt{\hat{\Sigma}_{t-1}^\theta} G_t$
8: **end for**
return Y_0

For DLIM, one can potentially provide Y_T as input, in order to use the support of the noise distribution as a latent space, alike what is done by Song et al. (2020). We give our DLIM sampling algorithm in Algorithm 4.

Algorithm 4 Deterministic sampling (DLIM)

Require: model $\{\hat{\epsilon}_t^\theta\}_{t=1}^T$, noise schedule $\{(\gamma_t, \sigma_t)\}_{t=1}^T$

1: Sample $Y_T \sim \mathcal{S}_\alpha^{\mathrm{i}}(0, \mathrm{I}_d)$
2: **for** $t \leftarrow T$ to 1 **do**
3: $Y_{t-1} \leftarrow \dfrac{Y_t}{\gamma_t} - \left(\dfrac{\sigma_{1\to t}}{\gamma_t} - \sigma_{1\to t-1} \right) \hat{\epsilon}_t^\theta(Y_t)$
4: **end for**
return Y_0

B Additional Remark on α-stable Distributions

Stable distributions are closed under convolution for a fixed value of α. Since convolution is equivalent to multiplication of the Fourier-transformed function, it follows that the product of two stable characteristic functions with the same α will yield another such characteristic function. We precisely characterize this stability property in the following proposition:

Proposition 3 (See (Nolan, 2020, Proposition 1.3))**.** *Let X_1, X_2 be two random variables respectively distributed as $X_1 \sim \mathcal{S}_{\alpha,\beta_1}(\mu_1, \sigma_1)$ and $X_2 \sim \mathcal{S}_{\alpha,\beta_2}(\mu_2, \sigma_2)$, with $\mu_1, \mu_2, \beta_1, \beta_2 \in \mathbb{R}$ and $\sigma_1, \sigma_2 > 0$. Then, $X = X_1 + X_2$ is distributed as $\mathcal{S}_{\alpha,\beta}(\mu, \sigma)$ where:*

$$\mu = \mu_1 + \mu_2 \ , \qquad \sigma = (\sigma_1^\alpha + \sigma_2^\alpha)^{1/\alpha} \ , \qquad \beta = \frac{\beta_1 \sigma_1^\alpha + \beta_2 \sigma_2^\alpha}{\sigma_1^\alpha + \sigma_2^\alpha} \ .$$

In particular, when X_1, X_2 are such that $X_1 \sim \mathcal{S}_\alpha(0, \sigma_1), X_2 \sim \mathcal{S}_\alpha(0, \sigma_2)$, then $X = X_1 + X_2 \sim \mathcal{S}_\alpha(0, (\sigma_1^\alpha + \sigma_2^\alpha)^{1/\alpha})$, which is the key relation used in the later characterizations of the distribution of our forward process.

C THEORETICAL DERIVATIONS FOR DLPM

In this section, we provide the detailed theory and associated derivations of the DLPM framework.

C.1 SETTING AND NOTATIONS

We will denote by $\phi_d(\cdot|\mu, \Sigma)$ the density of $\mathcal{N}(\mu, \Sigma)$, where $\mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d \times d}$. We will denote by $\psi^{(\alpha)}$ the density of $\mathcal{S}_{\alpha/2,1}(0, c_A)$, where $c_A = \cos^{2/\alpha}(\pi\alpha/4)$.

Forward process. We reintroduce the setting presented in Section 3, with the noising schedule being denoted by $\{(\gamma_t, \sigma_t)\}_{t=1}^T$, and the following **forward process** on which DLPM is based:

$$X_0 \sim p_\star \ , \quad \text{and} \quad X_t = \gamma_t X_{t-1} + \sigma_t \epsilon_t^{(\alpha)}, \tag{10}$$

where $p_\star$ is the data distribution and $\{\epsilon_t^{(\alpha)}\}_{t=1}^T$ are independent random variables distributed as $\mathcal{S}_\alpha^{\mathrm{i}}(0, \mathrm{I}_d)$.

Data augmentation process. We also introduce the associated **data augmentation process**:

$$Y_0 \sim p_\star \ , \quad \text{and} \quad Y_t = \gamma_t Y_{t-1} + \sigma_t A_t^{1/2} G_t \ , \tag{11}$$

where $\{G_t\}_{t=1}^T$ and $\{A_t\}_{t=1}^T$ are independent random variables distributed according to $G_t \sim \mathcal{N}(0, \mathrm{I}_d)$ and $A_t \sim \mathcal{S}_{\alpha/2,1}(0, c_A)$, with $c_A = \cos^{2/\alpha}(\pi\alpha/4)$. From Theorem 1, $\{Y_t\}_{t=0}^T$ is a Markov chain that admits the same distribution as $\{X_t\}_{t=0}^T$. We will denote by $p_t^{(\alpha)}$ the distribution of Y_t, and by $k_{t|t-1}^{(\alpha)}(\cdot|\cdot)$ the transition density associated to the Markov chain (10).

Backward process. A **backward process** associated to the Markov chain $\{Y_t\}_{t=0}^T$ is a Markov chain $\{\overleftarrow{Y'}_t\}_{t=0}^T$ such that the two processes $\{\overleftarrow{Y'}_{T-t}\}_{t=0}^T$ and $\{Y_t\}_{t=0}^T$ have the same distribution. For ease of presentation and following classical notations, we will rather consider $\{\overleftarrow{Y}_t\}_{t=0}^T$ where $\overleftarrow{Y}_t = \overleftarrow{Y'}_{T-t}$. We will denote by $\overleftarrow{k}_{t-1|t}^{(\alpha)}(\cdot|\cdot)$ the transition densities associated to the process $\{\overleftarrow{Y}_t\}_{t=0}^T$. Since the true backward process is never available to us, we will focus on an approximation induced by a variational family. We consider the process $\{\overleftarrow{Y}_t^\theta\}_{t=0}^T$ where $\overleftarrow{Y}_T^\theta$ is distributed as $\mathcal{S}(0, \sigma_{1\to t}\mathrm{I}_d)$, and the density of the distribution of $\overleftarrow{Y}_{t-1}^\theta$ conditioned on $\overleftarrow{Y}_t^\theta$ are given by $\overleftarrow{q}_{t-1|t}^\theta(\cdot|\cdot)$, where $\theta \in \mathbb{R}^D$ parameterizes a neural network. We also denote by $\overleftarrow{q}_{0:T}^\theta$ the joint distribution of $\{\overleftarrow{Y}^\theta\}_{t=0}^T$ and by $\overleftarrow{q}_t^\theta$ the marginal distribution of $\overleftarrow{Y}_t^\theta$.

Further notations. Finally, we denote by $p_t^\alpha(\cdot|a_{1:t})$, $k_{t|t-1,a}^{(\alpha)}(\cdot|\cdot, a_{1:t})$, and $\overleftarrow{q}_{t-1|t,a}^\theta(\cdot|\cdot, a_{1:t})$ the densities/transition densities associated to the processes $\{Y_t\}_{t=0}^T, \{\overleftarrow{Y}_t^\theta\}_{t=0}^T$ given $A_t = a_t$ for $1 \leqslant t \leqslant T$. We we will also write $p_t^\alpha(\cdot|y_0, a_{1:t})$, $k_{t|t-1,0,a}^{(\alpha)}(\cdot|\cdot, y_0, a_{1:t})$, and $\overleftarrow{q}_{t-1|t,0,a}^\theta(\cdot|\cdot, y_0, a_{1:t})$ when further conditioning on Y_0.

C.2 FORWARD PROCESS

Let us now characterize the distribution of X_t given X_0, and Y_t given $Y_0, \{A_t\}_{t=1}^T$, which are tractable thanks to Proposition 3. These will come in handy, for instance when working on the backward process in Appendix C.3.

Proposition 4 (Distribution of X_t given X_0)**.** *Let $\{X_t\}_{t=0}^T$ be the forward process as given in (10), and $\{(\gamma_t, \sigma_t)\}_{t=1}^T$ the noise schedule. Then the distribution of X_t given X_0 is given for any t by*

$$X_t \stackrel{d}{=} \gamma_{1\to t} X_0 + \sigma_{1\to t}\bar{\epsilon}_t$$

where $\bar{\epsilon}_t \sim \mathcal{S}_\alpha^i(0, \mathrm{I}_d)$, and $\gamma_{1\to t}, \sigma_{1\to t}$ are given by:

$$\gamma_{1\to t} = \prod_{k=1}^t \gamma_k \ , \qquad \sigma_{1\to t} = \left(\sum_{k=1}^t \left(\frac{\gamma_{1\to t}}{\gamma_{1\to k}} \sigma_k \right)^\alpha \right)^{1/\alpha} .$$

The proof is an elementary induction based on Proposition 3.

Proposition 5 (Distribution of Y_t given $Y_0, \{A_t\}_{t=1}^T$)**.** *Let $\{Y_t\}_{t=0}^T$ be the forward process as given in (11), $\{(\gamma_t, \sigma_t)\}_{t=1}^T$ the noise schedule, and $\{A_t\}_{t=1}^T$ the associated $\alpha/2$-stable random variables, parameterizing the variance of the Gaussian noise increments. Then the distribution of Y_t given $Y_0, \{A_t\}_{t=1}^T$ is the Gaussian distribution with mean $\gamma_{1\to t} Y_0$ and covariance matrix $\Sigma_{1\to t}\mathrm{I}_d$, i.e.,*

$$Y_t \stackrel{d}{=} \gamma_{1\to t} Y_0 + \Sigma_{1\to t}(A_{1:t})^{1/2} \bar{G}_t \ ,$$

where $\bar{G}_t \sim \mathcal{N}(0, \mathrm{I}_d)$, and

$$\gamma_{1\to t} = \prod_{k=1}^t \gamma_k \ , \qquad \Sigma_{1\to t}(a_{1:t}) = \sum_{k=1}^t \left(\frac{\gamma_{1\to t}}{\gamma_{1\to k}} \sqrt{a_k} \sigma_k \right)^2 . \tag{12}$$

The proof is elementary and omitted. It is worth mentioning the following recurrence, to speedup the computation of the sequence $\{\Sigma_{1\to t}(a_{1:t})\}_{t=1}^T$:

$$\Sigma_{1\to t}(a_{1:t}) = \sigma_t^2 a_t + \gamma_t^2 \Sigma_{1\to t-1}(a_{1:t-1}) \ .$$

C.3 BACKWARD PROCESS

Consider the setting of the data augmentation approach as given in (11). By the same arguments used in Section 3, it can be verified that a process starting from $p_T^{(\alpha)}$ and with transition densities $\overleftarrow{k}^{(\alpha)}_{t-1|t}(y_{t-1}|y_t) \propto p_t^{(\alpha)}(y_{t-1}) k^{(\alpha)}_{t|t-1}(y_t|y_{t-1})$ for any y_{t-1}, y_t is a backward process associated with $\{Y_t\}_{t=0}^T$. However, it raises two main problems. First (i), we cannot characterize the distribution of $\overleftarrow{k}^{(\alpha)}_{t-1|t}(y_{t-1}|y_t)$, since we do not know the data distribution. Second (ii), in the case where $\alpha < 2$, we do not have access to an explicit expression for $\overleftarrow{k}^{(\alpha)}_{t-1|t,0}(y_{t-1}|y_t, y_0)$.

Regarding (i), we have access to the distribution of Y_t given Y_0, so a valid strategy consists in devising a method relying on characterizing the backward of the process $\{Y_t\}_{t=0}^T$ given Y_0. This is the classical strategy used in DDPM (Ho et al. (2020)), which is possible in the case $\alpha = 2$ since $\overleftarrow{k}^{(\alpha)}_{t-1|t,0}(y_{t-1}|y_t, y_0)$ admits an analytical expression for any y_0, y_{t-1}, y_t, thanks to the properties of the Gaussian distribution.

Regarding (ii), in the case where $\alpha < 2$, we make use of the trick introduced in Theorem 1, justifying the data augmentation approach. We will rather characterize the density of the Markov kernels associated to the backward of the process $\{Y_t\}_{t=0}^T$ given Y_0 and $\{A_t\}_{t=1}^T$. This time, since we manage Gaussian noise increments, we can fall back to the classical strategy, as we further develop in the following proposition.

Proposition 6 (Density of the backward process associated to $\{Y_t\}_{t=0}^T$ given $Y_0, \{A_t\}_{t=1}^T$)**.** *Consider the setting of the data augmentation approach as given in (11). Let $\{(\gamma_t, \sigma_t)\}_{t=1}^T$ be the noise schedule at hand. Let $k^{(\alpha)}_{t-1|0,t,a}(\cdot|y_t, y_0, a_{1:t})$ be the density of the backward process associated to*

$\{Y_t\}_{t=0}^T$ *given* Y_0 *and* $\{A_t\}_{t=1}^T$. *Then* $k^{(\alpha)}_{t-1|t,0,a}(\cdot|y_t, y_0, a_{1:t})$ *is the density of a Gaussian distribution* $\mathcal{N}(\tilde{\mathtt{m}}_{t-1}, \tilde{\Sigma}_{t-1})$ *with mean* $\tilde{\mathtt{m}}_{t-1}$ *and variance* $\tilde{\Sigma}_{t-1}$ *such that*

$$\tilde{\mathtt{m}}_{t-1}(y_t, y_0, a_{1:t}) = \frac{1}{\gamma_t}\left(y_t - \Gamma_t(a_{1:t})\sigma_{1\to t}\epsilon_t(y_t, y_0)\right) \ , \quad \tilde{\Sigma}_{t-1}(a_{1:t}) = \Gamma_t(a_{1:t})\Sigma_{1\to t-1}(a_{1:t-1}) \ , \tag{13}$$

where

$$\Sigma_{1\to t}(a_{1:t}) = \sum_{k=1}^{t}\left(\frac{\gamma_{1\to t}}{\gamma_{1\to k}}\sqrt{a_k}\sigma_k\right)^2$$
$$\epsilon_t(y_t, y_0) = \frac{y_t - \gamma_{1\to t}y_0}{\sigma_{1\to t}}$$
$$\Gamma_t(a_{1:t}) = 1 - \frac{\gamma_t^2\Sigma_{1\to t-1}(a_{1:t-1})}{\Sigma_{1\to t}(a_{1:t})} \ .$$

Eventhough Γ_t *involves multiple heavy-tailed random variables, it is nonetheless bounded:* $0 \leqslant \Gamma_t \leqslant 1$.

Proof. To determine $k^{(\alpha)}_{t-1|t,0,a}(\cdot|y_t, y_0, a_{1:t})$, we need to work on the joint distribution of (Y_{t-1}, Y_t) conditioned on $Y_0, \{A_t\}_{t=1}^T$, which is a Gaussian vector for which classical techniques will let us derive the distribution of Y_{t-1} given Y_t. Before doing so we need to compute ρ_t the covariance of Y_{t-1} and Y_t given $Y_0, \{A_t\}_{t=1}^T$, which we do thanks to Proposition 5:

$$\rho_t = \mathrm{Cov}(Y_t, Y_{t-1}|Y_0, A_{1:T}) = \gamma_t\mathrm{Cov}(Y_{t-1}, Y_{t-1}|Y_0, A_{1:T}) = \gamma_t\Sigma_{1\to t-1}\mathrm{I}_d.$$

Denote by $k^{(\alpha)}_{t-1,t|0,a}$ the density of (Y_{t-1}, Y_t) conditioned on $Y_0, A_{1:T}$. Denote by $\phi_d(\cdot|\mu, \Sigma)$ the density of a d-dimensional Gaussian distribution with mean μ and covariance Σ. From the results of Proposition 5, we can write

$$k^{(\alpha)}_{t-1,t|0,a}(y_{t-1}, y_t|y_0, a_{1:t}) = \phi_d\left(\begin{pmatrix} y_{t-1} \\ y_t \end{pmatrix} \middle| \begin{pmatrix} \gamma_{1\to t-1}y_0 \\ \gamma_{1\to t}y_0 \end{pmatrix}, \begin{pmatrix} \Sigma_{1\to t-1}(a_{1:t-1})\mathrm{I}_d & \rho_t\mathrm{I}_d \\ \rho_t\mathrm{I}_d & \Sigma_{1\to t}(a_{1:t})\mathrm{I}_d \end{pmatrix}\right)$$

Then the distribution of Y_{t-1} given $Y_t, Y_0, A_{1:T}$ is a Gaussian distribution $\mathcal{N}(\tilde{\mathtt{m}}_{t-1}, \tilde{\Sigma}_{t-1})$ (Holt & Nguyen, 2023, Theorem 3) with mean $\tilde{\mathtt{m}}_{t-1}$ and variance $\tilde{\Sigma}_{t-1}$ satisfying:

$$\tilde{\mathtt{m}}_{t-1}(y_t, y_0, a_{1:t}) = \gamma_{1\to t-1}y_0 + \frac{\rho_t}{\Sigma_{1\to t}(a_{1:t})}\left(y_t - \gamma_{1\to t}y_0\right)$$
$$\tilde{\Sigma}_{t-1}(a_{1:t}) = \Sigma_{1\to t-1}(a_{1:t-1}) - \frac{\rho_t^2}{\Sigma_{1\to t}(a_{1:t})} \ .$$

By defining

$$\epsilon_t(y_t, y_0) = \frac{y_t - \gamma_{1\to t}y_0}{\sigma_{1\to t}} \ , \quad \Gamma_t(a_{1:t}) = 1 - \frac{\gamma_t^2\Sigma_{1\to t-1}(a_{1:t-1})}{\Sigma_{1\to t}(a_{1:t})} \ ,$$

we give the final expression for the mean $\tilde{\mathtt{m}}_{t-1}$ and variance $\tilde{\Sigma}_{t-1}$ of the distribution of Y_{t-1} given Y_t, Y_0 and $\{A_t\}_{t=1}^T$:

$$\tilde{\mathtt{m}}_{t-1}(y_t, y_0, a_{1:t}) = \frac{1}{\gamma_t}\left(y_t - \sigma_{1\to t}\Gamma_t(a_{1:t})\epsilon_t(y_t, y_0)\right) \tag{14}$$
$$\tilde{\Sigma}_{t-1}(a_{1:t}) = \Gamma_t(A_{1:t})\Sigma_{1\to t-1}(a_{1:t-1}) \ .$$

Since $\Gamma_t(a_{1:t}) = 1 - \gamma_t^2\Sigma_{1\to t-1}(a_{1:t-1})/\Sigma_{1\to t}(a_{1:t}) = a_t\sigma_t^2/\Sigma_{1\to t}(a_{1:t})$ and $a_t, \gamma_t, \Sigma_{1\to t}, \sigma_{1\to t} > 0$, we have $0 \leqslant \Gamma_t \leqslant 1$.

□

Case $\alpha = 2$ As we set $\alpha = 2$, the random variables $\{A_t\}_{t=1}^T$ become deterministic, equal to 2. One can check that in this case, with the variance preserving schedule

$$\gamma_t = \sqrt{1-\beta_t}\ , \quad \gamma_{1\to t} = \sqrt{\alpha_t}\ , \quad \sigma_t = \sqrt{\beta_t}\ , \quad \sigma_{1\to t} = \sqrt{1-\alpha_t}\ ,$$

then:

$$\Sigma_{1\to t} = 2\sigma_{1\to t}^2 \ = 2(1-\alpha_t)\ .$$

Further noticing that $\gamma_t = \gamma_{1\to t}/\gamma_{1\to t-1}$, one computes

$$\begin{aligned}\Gamma_t &= 1 - \frac{\sigma_{1\to t-1}^2 \alpha_t/\alpha_{t-1}}{\sigma_{1\to t}^2} \\ &= 1 - \frac{(1-\alpha_{t-1})\alpha_t/\alpha_{t-1}}{1-\alpha_t} \\ &= \frac{1-\alpha_t/\alpha_{t-1}}{1-\alpha_t}\ ,\end{aligned}$$

so that one recovers the famous equations made popular in the seminal DDPM paper (Ho et al., 2020, Equation 7):

$$\begin{aligned}\tilde{\mathtt{m}}_{t-1} &= \frac{\sqrt{\alpha_{t-1}}}{\sqrt{\alpha_t}}\left(Y_t - \frac{1-\alpha_t/\alpha_{t-1}}{\sqrt{1-\alpha_t}}\epsilon_t(Y_t, Y_0)\right) \\ \tilde{\Sigma}_{t-1} &= (1-\alpha_{t-1})\,\frac{1-\alpha_t/\alpha_{t-1}}{1-\alpha_t}\ ,\end{aligned} \tag{15}$$

with $\epsilon_t(Y_t, Y_0) = (Y_t - \sqrt{\alpha_t}Y_0)/\sqrt{1-\alpha_t}$.

Model for the reverse process. We propose approximating the backward process associated to $\{Y_t\}_{t=0}^T$, *given* $\{A_t\}_{t=1}^T$, adapting the DDPM approach. This time, for the backward process, we characterized the conditional density of Y_{t-1} given Y_0, Y_t and $\{A_t\}_{t=1}^T$:

$$\overleftarrow{k}^{(\alpha)}_{1:T|0,a}(y_{1:T}|y_0, a_{1:T}) := p_T^{(\alpha)}(y_T)\prod_{t=T}^{1} k^{(\alpha)}_{t-1|0,t,a}(y_{t-1}|y_t, y_0, a_{1:t})\ ,$$

where $k^{(\alpha)}_{t-1|0,t,a}(\cdot|y_t, y_0, a_{1:t})$ is the tractable density of a Gaussian distribution, as we have just proved in Proposition 6. We similarly reconsider the family of Gaussian variational approximation introduced in (4), modified to account for an i.i.d. sequence $\{A_t\}_{t=1}^T$:

$$\overleftarrow{q}^{\theta}_{t-1|t,a}(y_{t-1}|y_t, a_{1:t}) = \phi_d(y_{t-1}|\hat{\mathtt{m}}^{\theta}_{t-1}(y_t, a_{1:t}), \hat{\Sigma}^{\theta}_{t-1}(y_t, a_{1:t}))\ , \tag{16}$$

where ϕ_d is the density of the multivariate Gaussian distribution, so that the overall model for the backward process is the following:

$$\overleftarrow{q}^{\theta}_{0:T}(y_{0:T}) = \int \psi^{(\alpha)}_{1:T}(a_{1:T})p_T(y_T)\prod_{t=1}^{T}\overleftarrow{q}^{\theta}_{t-1|t,a}(y_{t-1}|y_t, a_{1:t})\mathrm{d}a_{1:T}\ ,$$

where $\psi^{(\alpha)}_{1:T}(a_{1:T}) = \prod_{i=1}^T \psi^{(\alpha)}(a_i)$.

C.4 Training Loss

In this section, we draw inspiration from DDPM (Ho et al. (2020)) to obtain a loss function admitting a closed-form formula. We further provide three design choices which lead to a simplified training loss, corresponding to the denoising loss for α-stable diffusion.

C.4.1 Classical loss for DDPM, $\alpha = 2$

We start by reviewing what is classically done for DDPM, *i.e.*, the case $\alpha = 2$. The variational approximation $\{\overleftarrow{q}^{\theta}_{0:T} : \theta \in \Theta\}$, for some parameter space Θ, is designed to admit the same decomposition as $\overleftarrow{k}^{(2)}_{0:T}$, *i.e.*, $\overleftarrow{q}^{\theta}_{0:T}(x_{0:T}) = \overleftarrow{q}^{\theta}_{T}(x_T)\prod_{t=T}^{1}\overleftarrow{q}^{\theta}_{t-1|t}(x_{t-1}|x_t)$, where $\overleftarrow{q}^{\theta}_{T}$ is chosen to be the density of $\mathcal{S}^{\mathrm{i}}_{\alpha}(0, \sigma_{1\to t}\mathrm{I}_d)$ as an approximation of $p_T^{(\alpha)}$. Then, it is trained on a classical upper bound of the KL loss $\mathscr{L}^{\mathrm{D}} : \theta \mapsto \mathrm{KL}(p_\star \| \overleftarrow{q}^{\theta}_{0})$ between the true and the generated distribution,

which is a form of evidence lower bound loss (Ho et al., 2020, Equation 5). Thus one resorts to optimize the following sum:

$$\mathscr{L}^{\mathrm{D}}(\theta) \leqslant \mathscr{L}^{\mathrm{D}}_T + \sum_{t=2}^{T} \mathscr{L}^{\mathrm{D}}_{t-1}(\theta) + \mathscr{L}^{\mathrm{D}}_0(\theta) + C \tag{17}$$

where C is a constant that does not depend on θ, and

$$\begin{aligned}
\mathscr{L}^{\mathrm{D}}_T &= \mathbb{E}\left[\mathrm{KL}\left(k^{(2)}_{T|0}(\cdot|X_0) \parallel \phi_d(\cdot|0, \sigma_{1\to T}\mathrm{I}_d)\right)\right] \\
\mathscr{L}^{\mathrm{D}}_0(\theta) &= -\mathbb{E}\left[\log\left(\overleftarrow{q}^{\theta}_{0|1}(X_0|X_1)\right)\right] \\
\mathscr{L}^{\mathrm{D}}_{t-1}(\theta) &= \mathbb{E}\left[\mathrm{KL}\left(k^{(2)}_{t-1|0,t}(\cdot|X_0, X_t) \parallel \overleftarrow{q}^{\theta}_{t-1|t}(\cdot|X_t)\right)\right] \ ,
\end{aligned}$$

where $\{X_t\}_{t=0}^T$ is the process defined in (10), and $k^{(2)}_{t-1|0,t}$ is the conditional density of X_{t-1} given X_0, X_t. We make the following classical remarks on the terms of this loss (Sohl-Dickstein et al. (2015), Yang et al. (2024)). The term $\mathscr{L}^{\mathrm{D}}_T$ does not depend on θ but only on the chosen time horizon for the forward process, that determines the final variance of the Gaussian distribution $\mathcal{N}(0, \sigma_{1\to T}\mathrm{I}_d)$. It is neglected. The effect of optimizing the first term $\mathscr{L}^{\mathrm{D}}_0(\theta)$ is negligible too.

More importantly, for the term $\mathscr{L}^{\mathrm{D}}_{t-1}(\theta)$, when using Gaussian variational approximations, *i.e.*, as

$$\overleftarrow{q}^{\theta}_{t-1|t}(x_{t-1}|x_t) = \phi_d\left(x_{t-1}|\hat{\mathtt{m}}^{\theta}_{t-1}(x_t), \hat{\Sigma}^{\theta}_{t-1}(x_t)\right) \ ,$$

where $(x, \mathtt{m}, \Sigma) \mapsto \phi_d(x|\mathtt{m}, \Sigma)$ is the d-dimensional density of the Gaussian distribution with mean $\mathtt{m}$ and covariance matrix Σ, $\hat{\mathtt{m}}^{\theta}_{t-1}, \hat{\Sigma}^{\theta}_{t-1}$ are some functions of x_t parameterized by θ, it turns out that $\mathscr{L}^{\mathrm{D}}_{t-1}$ admits a closed-form expression. For a fixed variance $\hat{\Sigma}^{\theta}_{t-1} = \tilde{\Sigma}_{t-1}$, with $\tilde{\Sigma}_{t-1}$ given in (15), one resorts to optimize a convenient $\mathbb{L}_2$ loss function:

$$\mathscr{L}^{\mathrm{D}}_{t-1}(\theta) = \lambda_t \|\tilde{\mathtt{m}}_{t-1}(x_t, x_0) - \hat{\mathtt{m}}^{\theta}_{t-1}(x_t)\|^2, \tag{18}$$

where $\lambda_t, \tilde{\mathtt{m}}_t$ depend on the noise schedule (γ_t, σ_t) and x_t, x_0.

Unfortunately, as we mentioned in Section 3.2.2, this solution cannot be used as such to learn the backward transitions associated to $\{X_t\}_{t=0}^T$ for $\alpha < 2$. The main issue that we face stems from the fact that the density of α-stable distributions are in most cases unknown, in contrast to Gaussian distributions. As a result, the conditional density $x_{t-1}, x_0, x_t \mapsto k^{(\alpha)}_{t-1|0,t}(x_{t-1}|x_0, x_t)$ is unknown for $\alpha < 2$, which prevents us to have an explicit expression for $\theta \mapsto \mathscr{L}^{\mathrm{D}}_{t-1}(\theta)$.

Moreover, the absence of a second order moment for α-stable distributions challenges the most straightforward adaptation we can make to the previous loss considering the data augmentation setting. Indeed, to fit θ to the data distribution, we aim to rely on Kullback-Leibler minimization, a.k.a. the maximum likelihood principle, and some associated upper bounds. A naive solution would consist in considering the bounds obtained applying the Jensen inequality:

$$\mathrm{KL}(p_\star \| \overleftarrow{q}^{\theta}_0) \leqslant \mathbb{E}\left[\mathrm{KL}\left[p_\star(\cdot) \| \overleftarrow{q}^{\theta}_{0|a}(\cdot|A_{1:T})\right]\right] \ ,$$

and fall back to the expression obtained in (17), only with conditioning on $\{A_t\}_{t=1}^T$. However, as we see in (18), this expression would involve taking expectation of A_t, while it is distributed as $\mathcal{S}_{\alpha/2,1}(0, c_A)$ and admits no first order moment. We are not aware of any bounds on $\mathrm{KL}(p_\star \parallel \overleftarrow{q}^{\theta}_{0|a}(\cdot|A_{1:T}))$ that would lead to a meaningful optimization problem due to the heavy tailed nature of the distribution of $\{A_t\}_{t=1}^T$.

C.4.2 Loss for DLPM, for any α

We now expose our methodology to address this limitation. We keep the structure of the classical loss and aim at minimizing the error between the backward process $\{\overleftarrow{Y}_t\}_{t=0}^T$ and its variational approximation $\{\overleftarrow{Y}^{\theta}_t\}_{t=0}^T$. To do so we consider the same loss structure as before, but take the square

root of individual KL terms. See Appendix C.4.3 for a more principled approach leading to a similar loss. Thus we consider the following valid loss function:

$$\mathscr{L}^{\mathrm{L}} : \theta \mapsto \mathscr{L}^{\mathrm{L}}(\theta) = \mathbb{E}\left[\sum_{t=2}^{T} \left(\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t})\right)^r\right] , \tag{19}$$

where $r > 0$,

$$\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t}) = \mathbb{E}\left[\mathrm{KL}\left(k^{(\alpha)}_{t-1|t,0,a}(\cdot|Y_t, Y_0, A_{1:t}) \parallel \overleftarrow{q}^{\theta}_{t-1|t,a}(\cdot|Y_t, A_{1:t})\right) \,\middle|\, A_{1:t}\right] ,$$

and $k^{(\alpha)}_{t-1|0,t,a}$ denotes the conditional density of Y_{t-1} given Y_0, Y_t and $\{A_t\}_{t=1}^T$. In order to ensure that the expectations with respect to $A_{1:T}$ are finite, we need to choose $r < \frac{\alpha}{2}$ when $\alpha \in (1, 2)$. For simplicity, in the rest of the paper, we will use $r = \frac{1}{2}$.

Proposition 7 (Training loss for DLPM)**.** *The loss $\mathscr{L}^{\mathrm{L}}$ admits a closed-form expression, such that one resorts to optimize the following loss for $1 \leqslant t \leqslant T$:*

$$\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t}) = \mathbb{E}\left[\frac{1}{2}\log\frac{\hat{\Sigma}^{\theta}_{t-1}}{\tilde{\Sigma}_{t-1}} + \frac{\tilde{\Sigma}_{t-1} + \|\tilde{\mathtt{m}}_{t-1} - \hat{\mathtt{m}}^{\theta}_{t-1}\|^2}{2\hat{\Sigma}^{\theta}_{t-1}} - \frac{1}{2} \,\middle|\, A_{1:t}\right]$$

where

$$\begin{aligned}
\tilde{\mathtt{m}}_{t-1}(Y_t, Y_0, A_{1:t}) &= \frac{1}{\gamma_t}\left(Y_t - \sigma_{1\to t}\Gamma_t(A_{1:t})\epsilon_t(Y_t, Y_0)\right) \\
\tilde{\Sigma}_{t-1}(A_{1:t}) &= \Gamma_t(A_{1:t})\Sigma_{1\to t-1}(A_{1:t-1}) \\
\epsilon_t(Y_t, Y_0) &= \frac{Y_t - \gamma_{1\to t}Y_0}{\sigma_{1\to t}} \\
\Sigma_{1\to t}(A_{1:t}) &= \sum_{k=1}^{t}\left(\frac{\gamma_{1\to t}}{\gamma_{1\to k}}\sqrt{A_k}\sigma_k\right)^2 \\
\Gamma_t(A_{1:t}) &= 1 - \frac{\gamma_t^2\Sigma_{1\to t-1}(A_{1:t-1})}{\Sigma_{1\to t}(A_{1:t})} ,
\end{aligned}$$

where $\hat{\mathtt{m}}^{\theta}_{t-1}, \hat{\Sigma}^{\theta}_{t-1}$ are the mean and variance of the backward transition kernels $\overleftarrow{q}^{\theta}_{t-1|t}$. We have omitted the arguments of the mean and variance functions for clarity.

Proof. Recall (Proposition 6) that the backward process Y_{t-1} conditioned on $\{A_t\}_{t=1}^T, Y_t, Y_0$ at time t is distributed as $\mathcal{N}(\tilde{\mathtt{m}}_{t-1}, \tilde{\Sigma}_{t-1})$, and, by design (Section 3.2.2), the backward transition kernels $\overleftarrow{q}^{\theta}_{t-1|t,a}$ of each element of the variational family describe a Gaussian transition kernel of mean $\hat{\mathtt{m}}^{\theta}_{t-1}$ and variance $\hat{\Sigma}^{\theta}_{t-1}$ at time t, as defined in 14. Since the KL term in $\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t})$ corresponds to a KL divergence between two Gaussian distributions, a closed-form formula is readily available. Here we rewrite the equation with all the functions arguments written out explicitly:

$$\begin{aligned}
\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t}) = \mathbb{E}\Bigg[&\frac{1}{2}\log\frac{\hat{\Sigma}^{\theta}_{t-1}(A_{1:t})}{\tilde{\Sigma}_{t-1}(A_{1:t})} - \frac{1}{2} \\
&+ \frac{\tilde{\Sigma}_{t-1}(A_{1:t}) + \|\tilde{\mathtt{m}}_{t-1}(Y_t, Y_0, A_{1:t}) - \hat{\mathtt{m}}^{\theta}_{t-1}(Y_t, A_{1:t})\|^2}{2\hat{\Sigma}^{\theta}_{t-1}(A_{1:t})} \,\Bigg|\, A_{1:t}\Bigg]
\end{aligned} \tag{20}$$

□

Now we discuss further design choices for $\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t})$, $\overleftarrow{q}^{\theta}_{t-1|t,a}$, leading to a simplified denoising loss, as is usually done in the literature. We denote them by **D1, D2** and **D3**.

D1. We set a fixed variance $\hat{\Sigma}^{\theta}_t = \tilde{\Sigma}_t$ for the reverse process , but we expect a study on the effect of learning variance to yield similar results as in original DDPM (Ho et al. (2020)), and especially its improved version (Nichol & Dhariwal (2021)).

D2. Following our own experimental results and the usual recommendation for denoising diffusion models (see, e.g., Yang et al. (2024); Karras et al. (2022)), we reparameterize the output of the model to predict the value $\epsilon_t(y_t, y_0)$ at time-step t rather than $\tilde{\mathtt{m}}_{t-1}(y_t, y_0, a_{1:t})$. Since

$$\tilde{\mathtt{m}}_{t-1}(Y_t, Y_0, A_{1:t}) = \frac{1}{\gamma_t}\left(Y_t - \sigma_{1\to t}\Gamma_t(A_{1:t})\epsilon_t(Y_t, Y_0)\right) \ ,$$

we re-parameterize $\hat{\mathtt{m}}^{\theta}_{t-1}$ to be equal to

$$\hat{\mathtt{m}}^{\theta}_{t-1}(Y_t, A_{1:t}) = \frac{1}{\gamma_t}\left(Y_t - \sigma_{1\to t}\Gamma_t(A_{1:t})\hat{\epsilon}^{\theta}_t(Y_t, A_{1:t})\right) \ .$$

with $\hat{\epsilon}^{\theta}_t$ being the output of the model. Following experimental results (see the introduction of Appendix G), we drop the dependency of $\hat{\epsilon}^{\theta}_t$ on $\{A_t\}_{t=1}^T$, which is reasonable since it approximates $\epsilon_t : (Y_t, Y_0) \mapsto \epsilon_t(Y_t, Y_0)$. Thus $\hat{\epsilon}^{\theta}_t$ only depends on t, Y_t. This choice achieves better performance in our experiments, allows further computational tricks (introduced in Appendix C.5), and enables one to re-use existing neural network architectures.

D3. Assuming D1, D2, $\mathscr{L}^{\mathrm{L}}_{t-1}(\theta)$ becomes

$$\mathscr{L}^{\mathrm{L}}_{t-1}(\theta) = \mathbb{E}\left[\lambda^2_{t,A_{1:t}}\|\hat{\epsilon}^{\theta}_t(Y_t, A_{1:t}) - \epsilon_t(Y_t, Y_0)\|^2\right] \ , \tag{21}$$

where $\lambda_{t,a_{1:t}} = \Gamma_t(a_{1:t})\sigma_{1\to t}/2\gamma_t\tilde{\Sigma}_{t-1}$, and $\epsilon_t(Y_t, Y_0) = (Y_t - \gamma_{1\to t}Y_0)/\sigma_{1\to t}$. The methodological knowledge of diffusion models motivates making specific choices for λ, different from its defined value, resulting in a classical technique for improving performances (e.g., see Karras et al. (2022); Ho et al. (2020); Nichol & Dhariwal (2021); Yang et al. (2024)). We will stick to the classical choice of choosing $\lambda_{t,a_{1:t}} = 1$, which experimentally works better and draws similarities to the score-based perspective (see Appendix E). Other choices and optimizations are left to further work.

The proof of Proposition 2 follows immediately from these design choices, hence omitted.

C.4.3 A PRINCIPLED APPROACH FOR DERIVING THE LOSS FUNCTION

In this section, we provide a more principled approach to derive the loss function for DLPM, as initially given in (19). We show the derivation for $r = 1/2$; however, the same derivation applies for any $r \in (0, 1]$.

Noting that Y_0 is independent of $\{A_t\}_{t=1}^T$ in (5), $p_\star$ is the equal to $k^{(\alpha)}_{0|a}(\cdot|a_{1:T})$, the conditional density of Y_0 given $A_t = a_t$ for any $t \in \{1, \dots, T\}$, and therefore, we consider the valid loss function

$$\mathscr{L}^{\mathrm{L}}(\theta) : \theta \mapsto \int \mathrm{d}a_{1:T}\psi^{(\alpha)}_{1:T}(a_{1:T})\left[\mathrm{KL}(k^{(\alpha)}_{0|a}(\cdot|a_{1:T})\|\overleftarrow{q}^{\theta}_{0|a}(\cdot|a_{1:T}))\right]^{1/2} \ .$$

While this function is still intractable, we can provide an upper bound which we can minimize. Indeed, using Jensen inequality twice, we bound this function by

$$\begin{aligned}
&\mathscr{L}^{\mathrm{L}}(\theta) = \int \mathrm{d}a_{1:T}\psi^{(\alpha)}_{1:T}(a_{1:T})\left\{\int \mathrm{d}y_0 k^{(\alpha)}_{0|a}(y_0|a_{1:T})(\log k^{(\alpha)}_{0|a}(y_0|a_{1:T}) - \log \overleftarrow{q}^{\theta}_{0|a}(y_0|a_{1:T}))\right\}^{1/2} \\
&\leqslant \int \mathrm{d}a_{1:T}\psi^{(\alpha)}_{1:T}(a_{1:T})\left\{-\int \mathrm{d}y_{0:T} k^{(\alpha)}_{0:T|0,a}(y_{0:T}|a_{1:T}) \log \frac{\overleftarrow{q}^{\theta}_{0:T|a}(y_{0:T}|a_{1:T})}{k^{(\alpha)}_{1:T|0,a}(y_{1:T}|y_0, a_{1:T})} + \mathrm{Cst}_1\right\}^{1/2} \\
&= \int \mathrm{d}a_{1:T}\psi^{(\alpha)}_{1:T}(a_{1:T})\{\textstyle\sum_{t=0}^{T-1}\mathscr{L}^{\mathrm{L}}_t(\theta, a_{1:T}) + \mathrm{Cst}_1 + \mathrm{Cst}_2\}^{1/2} \\
&\leqslant \int \mathrm{d}a_{1:T}\psi^{(\alpha)}_{1:T}(a_{1:T})\{\textstyle\sum_{t=0}^{T-1}\mathscr{L}^{\mathrm{L}}_t(\theta, a_{1:T})^{1/2} + \mathrm{Cst}_1^{1/2} + \mathrm{Cst}_2^{1/2}\} \ ,
\end{aligned}$$

where we used $\sqrt{a+b} < \sqrt{a} + \sqrt{b}$ when $a, b \geqslant 0$ and $\mathscr{L}^{\mathrm{L}}_0(\theta, a_{1:T}) = \mathbb{E}[-\log p_\theta(Y_0|Y_1, a_{1:T})|\{A_t\}_{t=1}^T = \{a_t\}_{t=1}^T]$ for $t > 0$,

$$\mathscr{L}^{\mathrm{L}}_t(\theta, a_{1:T}) = \mathbb{E}\left[\mathrm{KL}(k^{(\alpha)}_{t|t+1,0,a}(\cdot|Y_t, Y_0, a_{1:T})\|\overleftarrow{q}^{\theta}_{t|t+1,a}(\cdot|Y_t, a_{1:T}))|\{A_t\}_{t=1}^T = \{a_t\}_{t=1}^T\right] \ ,$$

and $\mathrm{Cst}_1 = \int \mathrm{d}y_0 p_\star(y_0)\log p_\star(y_0)\mathrm{d}y_0$ and $\mathrm{Cst}_2 = \mathbb{E}[\mathrm{KL}(k^{(\alpha)}_{1:T|0,a}(\cdot|Y_0, a_{1:T})|\overleftarrow{q}^{\theta}_T)|\{A_t\}_{t=1}^T = \{a_t\}_{t=1}^T]$ does not depend on θ since $\overleftarrow{q}^{\theta}_T$ is chosen as $\mathcal{S}_\alpha(0, \sigma_{1\to t}\mathrm{I}_d)$. Regarding $\mathscr{L}^{\mathrm{L}}_0$, we neglect this

term, replacing the distribution $\overleftarrow{q}^{\theta}_{0|1,a}(\cdot|y_1, a_{1:T})$ by a deterministic Dirac. One could alternatively employ the strategy of the discrete decoder for image data as described by Ho et al. (2020). We end up with the final loss function:

$$\mathscr{L}^{\mathrm{L}}(\theta) = \int \mathrm{d}a_{1:T} \psi^{(\alpha)}_{1:T}(a_{1:T}) \{\sum_{t=0}^{T-1} \mathscr{L}^{\mathrm{L}}_t(\theta, a_{1:T})^{1/2}\} \ .$$

We can then provide an explicit expression for $\mathscr{L}^{\mathrm{L}}_t(\theta, a_{1:T})$ based on the result of Proposition 6.

C.5 Reducing the computational cost with faster sampling at each timestep

In this section, we provide a faster algorithm for training DLPM, computing each loss term $\mathscr{L}^{\mathrm{L}}_{t-1}$ using only two heavy-tailed random variables per datapoint, instead of t random variables.

Consider again the process $\{Y_t\}_{t=0}^T$. We replace the loss function (19) by an equivalent one:

$$\mathscr{L}^{\mathrm{L}}_{\text{time}}(\theta) = \mathbb{E}\left[\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t})^{1/2}\right] \ , \tag{22}$$

where $t \sim \mathcal{U}[2, T] = (\sum_{i=2}^T \delta_i)/(T-1)$. The standard technique for computing the loss consists in the following loop:

1. Take a batch of B datapoints $\{Y_0^i\}_{i=1}^B$.
2. For each datapoint Y_0^i, draw a random t_i,as suggested by the alternative loss (22). Indeed, for a single datapoint, (i) training on all timesteps rather than just one yields equal to inferior results, for a much higher computational cost, (ii) it is beneficial for the model to proportionally spend more time learning specific time ranges (iii) thus the distribution of t can be optimized and is a matter of ongoing methodological research, e.g., see Karras et al. (2022).
3. Draw sequences $\{A_t^i\}_{t=1}^{t_i}$ of heavy-tailed random variables.
4. Compute the noised datapoints $\{Y_{t_i}^i\}_{i=1}^B$.
5. Compute the batch loss

$$\hat{\mathscr{L}}^{\mathrm{L}}(\theta) = \frac{1}{B}\sum_{i=1}^B \mathscr{L}^{\mathrm{L}}_{t_i-1}(\theta, Y^i_{t_i}, A^i_{1:t_i}) \ ,$$

   such that $\hat{\mathscr{L}}^{\mathrm{L}}(\theta) \approx \mathscr{L}^{\mathrm{L}}_{\text{time}}(\theta)$.
6. Do an optimization step.

Step 3 can be expensive, since one has to sample on average $O(T)$ d-dimensional heavy-tailed random variables to compute a single noised datapoint Y_t from Y_0. This is all the more inefficient as T can be quite large (indeed, on image datasets we can have $T = 4000$, see Appendix G).

One can guess that this is abusive, especially since characterizing the distribution of Y_t given Y_0 only requires a single heavy-tailed random variable:

$$Y_t \stackrel{d}{=} \gamma_{1\to t} Y_0 + \sigma_{1\to t} \bar{A}^{1/2} \bar{G}_t \ ,$$

where $\bar{A} \sim \mathcal{S}_{\alpha/2,1}(0, c_A)$, $\bar{G}_t \sim \mathcal{N}(0, \mathrm{I}_d)$. As we formalize in the next proposition, it is indeed possible to bypass the sampling of a whole sequence. Since we manipulate the joint distribution of (Y_0, Y_{t-1}, Y_t) for the loss term $\mathscr{L}^{\mathrm{L}}_{t-1}(\theta)$, we will actually need to sample two heavy-tailed random variables.

Proposition 8 (Sampling two heavy-tailed r.v for each loss term)**.** *Suppose that the functions* $\hat{\mathtt{m}}^{\theta}_{t-1}, \hat{\Sigma}^{\theta}_{t-1}$ *satisfy for any* $y_t, a_{1:t}$*:*

$$\hat{\mathtt{m}}^{\theta}_{t-1}(y_t, a_{1:t}) = M^{\theta}_{t-1}\left(y_t, a_t, \frac{\Sigma_{1\to t-1}(a_{1:t-1})}{\sigma^2_{1\to t-1}}\right)$$
$$\hat{\Sigma}^{\theta}_{t-1} = S^{\theta}_{t-1}\left(y_t, a_t, \frac{\Sigma_{1\to t-1}(a_{1:t-1})}{\sigma^2_{1\to t-1}}\right) \ ,$$

for some functions $M^\theta_{t-1}, S^\theta_{t-1}$, and where

$$\Sigma_{1\to t}(a_{1:t}) = \sum_{k=1}^{t} \left(\frac{\gamma_{1\to t}}{\gamma_{1\to k}} \sqrt{a_k}\sigma_k \right)^2 ,$$

as given in (12). Then each term $\mathbb{E}[\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t})]^{1/2}$ of the loss can be computed sampling only two independent random variables $\bar{A}^t_0, \bar{A}^t_1$ distributed as $\mathcal{S}_{\alpha/2,1}(0, c_A)$:

$$\mathbb{E}\left[\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t})\right]^{1/2} = \mathbb{E}\left[\mathscr{L}^{\mathrm{Less}}_{t-1}(\theta, \bar{A}^t_{0,1})\right]^{1/2} ,$$

where $\bar{A}^t_{0,1} := (\bar{A}^t_0, \bar{A}^t_1)$, and

$$\mathscr{L}^{\mathrm{Less}}_{t-1}(\theta, \bar{A}^t_{0,1}) = \mathbb{E}\Bigg[\frac{1}{2}\log\frac{S^\theta_{t-1}(Z_t, \bar{A}^t_{0,1})}{\tilde{\Sigma}_{t-1}(\bar{A}^t_{0,1})} - \frac{1}{2} + \frac{\tilde{\Sigma}_{t-1}(\bar{A}^t_{0,1}) + \|\tilde{\mathtt{m}}'_{t-1}(Z_t, Z_0, \bar{A}^t_{0,1}) - M^\theta_{t-1}(Z_t, \bar{A}^t_{0,1})\|^2}{2S^\theta_{t-1}(Z_t, \bar{A}^t_{0,1})} \,\Bigg|\, \bar{A}^t_{0,1}\Bigg] , \tag{23}$$

with $\{Z_t\}_{t=0}^T$ being a stochastic process defined as

$$Z_0 = Y_0 , \quad Z_t = \gamma_{1\to t} Z_0 + \Sigma'^{1/2}_t \bar{G}_t ,$$

where $\{G_t\}_{t=1}^T$ is an i.i.d. sequence distributed as $\mathcal{N}(0, \mathrm{I}_d)$, and

$$\begin{aligned}
\Sigma'_{t-1}(\bar{A}^t_0) &= \sigma^2_{1\to t-1}\bar{A}^t_0 \\
\Sigma'_t(\bar{A}^t_{0,1}) &= \sigma^2_t \bar{A}^t_1 + \gamma^2_t \Sigma'_{t-1}(\bar{A}^t_0) \\
\Gamma'_t(\bar{A}^t_{0,1}) &= 1 - \frac{\gamma^2_t \Sigma'_{t-1}(\bar{A}^t_0)}{\Sigma'_t(\bar{A}^t_{0,1})} ,
\end{aligned}$$

such that $Z_t \overset{d}{=} Y_t$, and:

$$\begin{aligned}
\tilde{\mathtt{m}}_{t-1}(Z_t, Z_0, \bar{A}^t_{0,1}) &= \frac{1}{\gamma_t}\left(Z_t - \sigma_{1\to t}\Gamma'_t(\bar{A}^t_{0,1})\epsilon_t(Z_t, Z_0)\right) \\
\tilde{\Sigma}_{t-1}(\bar{A}^t_{0,1}) &= \Gamma'_t(\bar{A}^t_{0,1})\Sigma'_{t-1}(\bar{A}^t_0) \\
\epsilon_t(Z_t, Z_0) &= \frac{Z_t - \gamma_{1\to t} Z_0}{\sigma_{1\to t}} .
\end{aligned}$$

In order to keep the notations similar for all $t \geqslant 2$, in the case of $\mathscr{L}^{\mathrm{Less}}_1$, we always set $\bar{A}^2_0 = 0$.

Proof. Remember the full equation for the loss, first given in Proposition 7 (20):

$$\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t}) = \mathbb{E}\Bigg[\frac{1}{2}\log\frac{\hat{\Sigma}^\theta_{t-1}(A_{1:t})}{\tilde{\Sigma}_{t-1}(A_{1:t})} - \frac{1}{2} + \frac{\tilde{\Sigma}_{t-1}(A_{1:T}) + \|\tilde{\mathtt{m}}_{t-1}(Y_t, Y_0, A_{1:T}) - \hat{\mathtt{m}}^\theta_{t-1}(Y_t, A_{1:T})\|^2}{2\hat{\Sigma}^\theta_{t-1}(A_{1:t})} \,\Bigg|\, A_{1:t}\Bigg] .$$

Now, all the required variables and computations only depend on $A_t, \Sigma_{1\to t-1}$; this is the case for $\hat{\mathtt{m}}^\theta_{t-1}, \hat{\Sigma}^\theta_{t-1}$ by hypothesis, and this is the case for $\tilde{\mathtt{m}}_{t-1}, \tilde{\Sigma}_{t-1}$ as one can see in (13). Rewriting the previous loss as

$$\begin{aligned}
\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t}) = \mathbb{E}\Bigg[&\frac{1}{2}\log\frac{S^\theta_{t-1}\left(Z_t, A_t, \frac{\Sigma_{1\to t-1}(A_{1:t-1})}{\sigma^2_{1\to t-1}}\right)}{\tilde{\Sigma}_{t-1}(A_{1:T})} - \frac{1}{2} \\
&+ \frac{\tilde{\Sigma}_{t-1}(A_{1:t})}{2S^\theta_{t-1}\left(Z_t, A_t, \frac{\Sigma_{1\to t-1}(A_{1:t-1})}{\sigma^2_{1\to t-1}}\right)} \\
&+ \frac{\|\tilde{\mathtt{m}}_{t-1}(Y_t, Y_0, A_{1:t}) - M^\theta_{t-1}\left(Z_t, A_t, \frac{\Sigma_{1\to t-1}(A_{1:t-1})}{\sigma^2_{1\to t-1}}\right)\|^2}{2S^\theta_{t-1}\left(Z_t, A_t, \frac{\Sigma_{1\to t-1}(A_{1:t-1})}{\sigma^2_{1\to t-1}}\right)} \,\Bigg|\, A_t, \Sigma_{1\to t-1}\Bigg] ,
\end{aligned}$$

it becomes clear how the expectation can be taken on the joint distribution of

$$\left(Y_0, Y_{t-1}, Y_t, A_t, \frac{\Sigma_{1\to t-1}(A_{1:t-1})}{\sigma^2_{1\to t-1}}\right) .$$

A direct application of Lemma 1 shows that this expectation can be taken on the joint distribution of the five random variables $(Z_0, Z_{t-1}, Z_t, \bar{A}_1^t, \bar{A}_0^t)$, which only necessitates sampling two heavy-tailed random variables $\bar{A}_0^t, \bar{A}_1^t$. Using the formulas for Z_0, Z_{t-1} and Z_t given $\bar{A}_0^t, \bar{A}_1^t$ as defined in Lemma 1, we obtain the equivalent loss (23).

□

As we will prove in the next proposition, the conditions of Proposition 8 are always satisfied under design choices D1, D2. Under design choice D3, we can also rewrite the simplified denoising loss given in Proposition 2.

Proposition 9 (Sampling one heavy-tailed r.v in the simplified loss)**.** *Assume the design choices D1, D2, D3 are satisfied. Then one can obtain the following simplified denoising objective function from the full objective function given in* (23)*:*

$$\mathscr{L}^{\text{SimpleLess}}(\theta) = \sum_{t=1}^{T} \mathbb{E}\left[\mathbb{E}\left(\|\hat{\epsilon}_t^\theta(Z_t) - \bar{A}_t^{1/2}\bar{G}_t\|^2 \,|\bar{A}_t\right)^{1/2}\right] ,$$

where $\{\bar{A}_t\}_{t=1}^T$ *is an i.i.d. sequence distributed as* $\mathcal{S}_{\alpha/2,1}(0, c_A)$*, and*

$$Z_t = \gamma_{1\to t} Z_0 + \sigma_{1\to t}\bar{A}_t^{1/2}\bar{G}_t ,$$

with $\{\bar{G}_t\}_{t=1}^T$ *an i.i.d. sequence distributed as* $\mathcal{N}(0, \mathrm{I}_d)$*.*

Proof. Let us recall design choice D1:

$$\hat{\Sigma}_{t-1}^\theta(A_{1:}) = \Gamma_t(A_{1:t})\Sigma_{1\to t}(A_{1:t}) ,$$

and design choice D2:

$$\hat{\mathtt{m}}_{t-1}^\theta(Z_t, A_{1:t}) = \frac{1}{\gamma_t}\left(Z_t - \sigma_{1\to t}\Gamma_t(A_{1:t})\hat{\epsilon}_t^\theta(Z_t)\right) .$$

Since Γ only depends on $\Sigma_{1\to t}$ and $\Sigma_{1\to t-1}$, both $\hat{\mathtt{m}}_{t-1}^\theta, \tilde{\Sigma}_{t-1}$ can be expressed as functions of z_t, a_t and $\Sigma_{1\to t}(a_{1:t})$. Thus the assumptions of Proposition 8 are satisfied. Using the same notations, we can apply the same algebraic transformations as in (21), and by design choice D3, obtain:

$$\mathscr{L}^{\text{Less}}(\theta) = \sum_{t=1}^{T} \mathbb{E}\left[\mathbb{E}\left(\|\hat{\epsilon}_t^\theta(Z_t) - \epsilon_t(Z_t, Z_0)\|^2 \,|\bar{A}_0^t, \bar{A}_1^t\right)^{1/2}\right] .$$

Finally, we apply Lemma 2 to Σ_t' to affirm that $\Sigma_t' \overset{d}{=} \sigma^2_{1\to t}\bar{A}_t$, where $\bar{A}_t \sim \mathcal{S}_{\alpha/2,1}(0, c_A)$, and obtain the final loss we presented. □

We stress that this denoising training loss is similar to that of LIM (Yoon et al., 2023, Theorem 4.3), but elevated to the necessary power to guarantee that the loss is finite. See Appendix E.1 for a more detailed discussion.

D Denoising Lévy Implicit Models (DLIM)

Using the same techniques as in DDIM (Song et al. (2020)), we obtain a deterministic sampling process which we naturally call Denoising Levy Implicit Models (DLIM). Alike the Gaussian case treated in the original DDIM work, we will show that both DLPM and DLIM can share the same neural network.

D.1 Non-Markovian forward process

Let $\{\rho_t\}_{t=1}^T$ be an alternative noise schedule, proper to DLIM, that will ultimately tend to zero for deterministic generation. In the same way as in Section 3.2.2, we take a data augmentation approach. We consider a process $\{Z_t\}_{t=1}^T$ defined by $Z_0 \sim p_\star$, where $p_\star$ is the data distribution, $Z_T \sim \mathcal{S}(\gamma_{1\to T} Z_0, \sigma_{1\to T}\mathrm{I}_d)$ and, for $1 < t \leqslant T$

$$Z_{t-1} = \gamma_{1\to t-1} Z_0 + (\sigma_{1\to t-1}^\alpha - \rho_t^\alpha)^{1/\alpha} \epsilon_t(Z_t, Z_0) + \rho_t A_t^{1/2} G_t$$

where $\epsilon_t(Z_t, Z_0) = (Z_t - \gamma_{1\to t} Z_0)/\sigma_{1\to t}$, and $\{A_t\}_{t=1}^T$, $\{G_t\}_{t=1}^T$ are independent random variables distributed according to $A_t \sim \mathcal{S}_{\alpha/2,1}(0, c_A)$ and $G_t \sim \mathcal{N}(0, \mathrm{I}_d)$.

Proposition 10. *The distribution of Z_t given Z_0 is the same as that of Y_t given Y_0.*

Proof. This is a simple proof by induction, where one can re-adapt the technique of (Song et al., 2020, Lemma B.1) with the property for addition of α-stable variable as we introduced in Proposition 3. The case $t = T$ is true by construction. Suppose now that the property is verified at timestep t, where $1 \leqslant t \leqslant T$. Then, focusing on the distribution of Z_{t-1} given Z_0, $\epsilon_t(Z_t, Z_0) = (Z_t - \gamma_{1\to t} Z_0)/\sigma_{1\to t}$ is distributed as $\mathcal{S}_\alpha^{\mathrm{i}}(0, \mathrm{I}_d)$ by hypothesis, and thus by Proposition 3 and since $A_t^{1/2} G_t \sim \mathcal{S}_\alpha^{\mathrm{i}}(0, \mathrm{I}_d)$, we can write

$$Z_{t-1} \overset{d}{=} \gamma_{1\to t-1} Z_0 + \sigma_{1\to t-1} \bar{\epsilon}_t \ ,$$

where $\bar{\epsilon}_t \sim \mathcal{S}_\alpha^{\mathrm{i}}(0, \mathrm{I}_d)$, which shows that indeed Z_{t-1} given Z_0 admits the same distribution as Y_{t-1} given Y_0. □

The design of this process makes the distribution of Z_t given Z_0 match that of Y_t given Y_0, where $\{Y_t\}_{t=0}^T$ is the forward process of DLPM (5). The task of sampling from it is thus efficient and straightforward.

D.2 Generative process

We similarly reconsider the family of Gaussian variational approximation introduced in (4), which accounts for an i.i.d. sequence $\{A_t\}_{t=1}^T$:

$$\overleftarrow{q}^\theta_{0:T}(y_{0:T}) = \int \psi^{(\alpha)}_{1:T}(a_{1:T}) p_T(y_T) \prod_{t=1}^T \overleftarrow{q}^\theta_{t-1|t,a}(y_{t-1}|y_t, a_{1:t}) \mathrm{d}a_{1:T} \ ,$$

where $\psi^{(\alpha)}_{1:T}(a_{1:T}) = \prod_{i=1}^T \psi^{(\alpha)}(a_i)$, $\psi^{(\alpha)}$ is the density of $\mathcal{S}_{\alpha/2,1}(0, c_A)$, and p_T is the density of $\mathcal{S}(0, \sigma_{1\to T}\mathrm{I}_d)$. This time, we set

$$\overleftarrow{q}^\theta_{t-1|t,a}(z_{t-1}|z_t, a_{1:t}) = \phi_d\left(z_{t-1}|\hat{\mathtt{m}}^\theta_{t-1}(z_t),\ \rho_t^2 a_t\right) \ ,$$

with ϕ_d being the density of the multivariate Gaussian distribution: the variance is fixed, determined by the alternative noise schedule $\{\rho_t\}_{t=1}^T$. For deterministic sampling, one will ultimately choose $\rho_t = 0$ for all t and sample the chain $\{\overleftarrow{Z}^\theta_t\}_{t=0}^T$ as follows:

$$\overleftarrow{Z}^\theta_T \sim \mathcal{S}(0, \sigma_{1\to T}\mathrm{I}_d) \ , \quad \overleftarrow{Z}^\theta_{t-1} = \hat{\mathtt{m}}^\theta_t(\overleftarrow{Z}^\theta_t) \text{ for } t \in \{T, \cdots, 1\} \ .$$

As z_t is available as input, the model can be fit to approximate the value of $\epsilon_t(z_t, z_0)$, and we reparameterize $\hat{\mathtt{m}}^\theta_t$ as follows:

$$\hat{\mathtt{m}}^\theta_t(z_t) = \frac{z_t - \sigma_{1\to t}\hat{\epsilon}^\theta_t(z_t)}{\gamma_t} + (\sigma_{1\to t-1}^\alpha - \rho_t^\alpha)^{1/\alpha} \hat{\epsilon}^\theta_t(z_t) \ . \tag{24}$$

This is alike design choice D2 in Appendix C.4.2.

D.3 Loss function and equivalence with DLPM

We denote by $h^{(\alpha)}_{t-1|t,0,a}$ the density of Z_{t-1} given Z_t, Z_0 and $A_{1:T}$, which is the density of the Gaussian distribution with mean $\gamma_{1\to t-1}Z_0 + (\sigma^\alpha_{1\to t-1} - \rho^\alpha_t)^{1/\alpha}\epsilon_t(Z_t, Z_0)$ and covariance $\rho^2_t A_t \mathrm{I}_d$. Since this distribution is now a given, we are inclined to use the loss function introduced in (7), which is:

$$\mathscr{L}^{\mathrm{L}}(\theta) := \mathbb{E}\left[\sum_{t=1}^{T}\left(\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t})\right)^{1/2}\right] , \qquad \text{where}$$

$$\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t}) := \mathbb{E}\left[\mathrm{KL}\left(h^{(\alpha)}_{t-1|t,0,a_{1:t}}(\cdot|Z_t, Z_0, A_{1:t}) \parallel \overleftarrow{q}^{\theta}_{t-1|t,a}(\cdot|Z_t, A_{1:t})\right)\Big| A_{1:T}\right] .$$

Since for $1 \leqslant t \leqslant T$, $h^{(\alpha)}_{t-1|t,0,a}$ and $\overleftarrow{q}^{\theta}_{t-1|t,a}$ are the densities of Gaussian distributions, we can analytically compute each term of the loss, as in (19):

$$\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t}) = \frac{1}{2\rho^2_t A_t}\|\gamma_{1\to t-1}Z_0 + (\sigma^\alpha_{1\to t-1} - \rho^\alpha_t)^{1/\alpha}\epsilon_t(Z_t, Z_0) \; - \hat{\mathtt{m}}^{\theta}_t(Z_t, A_{1:t})\|^2$$

where $\epsilon_t(Z_t, Z_0) = (Z_t - \gamma_{1\to t}Z_0)/\sigma_{1\to t}$. Since the variance of the elements of our variational family $\{\overleftarrow{q}^{\theta}_{0:T}\}$ have been designed to match that of the backward process given $Z_0, A_{1:T}$, the expression for the loss is readily in a simpler format. Finally, using the reparameterization given in (24), The loss term $\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:T})$ becomes:

$$\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:t}) = \lambda'_{t,A_t}\|\epsilon_t(Z_t, Z_0) - \hat{\epsilon}^{\theta}_t(Z_t)\|^2 ,$$

where $\lambda'_{t,a_t} = (\sigma_{1\to t} - (\sigma^\alpha_{1\to t-1} - \rho^\alpha_t)^{1/\alpha})^2/(2\rho^2_t a_t)$. By comparing with the simpler DLPM loss (21) with design choices D1, D2, as introduced in Appendix C.4.2, we realize we obtained the same loss term, with a different multiplicative factor λ'_{t,a_t} instead of $\lambda_{t,a_{1:t}}$ in (21). Finally, considering the alternative loss where $\lambda'_{t,a_t} = 1$ for all t, alike the design choice D3 in Appendix C.4.2, we fall back to the same simplified objective function obtained for DLPM:

$$\mathscr{L}^{\mathrm{Simple}}(\theta) = \sum_{t=1}^{T}\mathbb{E}\left[\mathbb{E}\left(\|\hat{\epsilon}^{\theta}_t(Y_t) - \epsilon_t(Y_t, Y_0)\|^2 \mid A_{1:t}\right)^{1/2}\right] ,$$

E Additional Information on Levy-Ito Models (LIM)

Here we briefly recapitulate the work done by Yoon et al. (2023), introducing continuous diffusion models with α-stable heavy-tailed noise. Using notations closer to ours, we define the noising schedule as any locally bounded continuous functions $\gamma : (t, X) \mapsto \gamma(t, X)$ and $\sigma : (t) \mapsto \sigma(t)$. We denote by L^α_t the Levy process for which the increments between time $s < t$ follow a symmetric isotropic α-stable distribution $\mathcal{S}^{\mathrm{i}}_\alpha(0, (t-s)\mathrm{I}_d)$. In this setting, the forward process X_t, with $X_0 \sim p_\star$, is written

$$\mathrm{d}X_t = \gamma(t, X_{t-})\mathrm{d}t + \sigma(t)\mathrm{d}L^\alpha_t , \tag{25}$$

where X_{t-} denotes the left limit of X at time t. Similarly, X_t is distributed as $\mathcal{S}^{\mathrm{i}}_\alpha(\gamma_{1\to t}X_0, \sigma_{1\to t}\mathrm{I}_d)$ when using Euler steps. This defines the cadlag (right continuous with left limits) solution, which in the case of $\alpha < 2$ a.s admits discontinuous jumps. We then consider the following backward process $\overleftarrow{X}_t$:

$$\mathrm{d}\overleftarrow{X}_t = \left(-\gamma(t, \overleftarrow{X}_{t+}) - \alpha\sigma^\alpha(t, \overleftarrow{X}_{t+})S^{(\alpha)}_t(\overleftarrow{X}_{t+})\right)dt + \sigma(t)\mathrm{d}\bar{L}^\alpha{}_t + d\bar{Z}_t \tag{26}$$

where $\bar{Z}_t$ is the backward version of a Levy-type stochastic integral Z_t s.t $\mathbb{E}[Z_t] = 0$ with finite variation, and $S^{(\alpha)}_t$ is the fractional score function, defined to be

$$S^{(\alpha)}_t(x) = \frac{\Delta^{\frac{\alpha-2}{2}}\nabla p_t(x)}{p_t(x)} ,$$

where $\Delta^{\eta/2}$ denotes the fractional Laplacian of order $\eta/2$ (Ortigueira et al. (2014)). More precisely, $\Delta^{\eta/2}f(x) = \mathcal{F}^{-1}\{\|u\|^\eta\mathcal{F}\{f\}(u)\}$, where $\mathcal{F}, \mathcal{F}^{-1}$ are the Fourier and inverse Fourier transforms.

The training loss is obtained using the classical technique of denoising score matching (Vincent (2011)), where the following losses

$$L : \theta \mapsto \mathbb{E}\|s_\theta(X_t, t) - S_t^{(\alpha)}(X_t)\|^2 \ , \qquad L' : \theta \mapsto \mathbb{E}\|s_\theta(X_t, t) - S_t^{(\alpha)}(X_t|X_0)\|^2 \ , \tag{27}$$

are proven to be equivalent objective functions, with s_θ being the score approximation given by the model.

E.1 Comparing LIM and DLPM

Let $(X_t)_{0\leqslant t\leqslant T}$ be the forward process introduced in (25). As stressed initially, the framework of LIM is not straightforward to manipulate, thus we do not characterize explicitly the distribution of X_t given X_0 for an arbitrary noise schedule in the continuous case. Since the work done for LIM by Yoon et al. (2023) only provides the formulas for the scale-preserving schedule, we stick to them in the following: we keep the notation $\gamma_{1\to t}, \sigma_{1\to t}$ for the continuous time regime equivalent of the scale preserving schedule we introduce in Appendix G, and they match on integer times t.

Considering an Euler scheme to obtain discretization for the forward and backward process, and using our own notations, both LIM and DLPM admit the same forward process $\{X_t\}_{t=1}^T$, $X_0 \sim p_\star$ and

$$X_t = \gamma_t X_{t-1} + \sigma_t \epsilon_t^{(\alpha)},$$

where $\{\epsilon_t^{(\alpha)}\}_{t=1}^T$ is an iid sequence of random variable distributed as $\mathcal{S}_\alpha^{\mathrm{i}}(0, \mathrm{I}_d)$. We denote by $\{\overleftarrow{X}_t^\theta\}_{t=T}^0$ the backward process associated to the Euler discretization of (26), where we use a neural network s_θ to approximate the true score $S_t^{(\alpha)}$. Since the true score of the data $S_t^{(\alpha)}(x_t|x_0)$ can be expressed as

$$S_t^{(\alpha)}(x_t|x_0) = -\frac{1}{\alpha\sigma_{1\to t}^{\alpha-1}}\epsilon_t(x_t, x_0) \ ,$$

where $\epsilon_t(x_t, x_0) = (x_t - \gamma_{1\to t}x_0)/\sigma_{1\to t}$, we write

$$s_\theta(x_t, t) = -\frac{1}{\alpha\sigma_{1\to t}^{\alpha-1}}\hat{\epsilon}_t^\theta(x_t, x_0) \ ,$$

so that we rather work with $\hat{\epsilon}_t^\theta$, with the same intention that led us to the design choices given in Appendix C.4.2.

Moreover, we denote by $\{\overleftarrow{Y}_t^\theta\}_{t=0}^T$ the backward process of DLPM, as introduced in (16). As emphasized in Table 4, the sampling strategies for LIM and DLPM differ fundamentally when $\alpha \neq 2$. This is also the case for the training procedure.

Stochastic sampling. The DLPM approach introduces the bounded random variable $0 \leqslant \Gamma_t \leqslant 1$, interacting with the mean and variance of the Gaussian conditional at hand. Three points: when $\alpha = 2$, Γ_t becomes deterministic and one recovers DDPM formulas. Second, Γ_t brings additional stochasticity in the sampling process. Third, it does so in the interesting manner than it simultaneously scales both (i) the magnitude of the noise added at time $t-1$ and (ii) the output of the noise model.

Deterministic sampling. In the case of the scale-preserving schedule, these two equations do not describe the same sampling procedure.

Training. Alike the Gaussian case ($\alpha = 2$), the score $S_t^{(\alpha)}(x_t|x_0)$ is a linear expression of the noise term $\epsilon_t(x_t, x_0)$, so the training equations are very similar, and can be reformulated to involve a denoising loss:

$$\mathcal{L}_{t-1} : \theta \mapsto \mathbb{E}\left(\|\hat{\epsilon}_t^\theta(X_t) - \epsilon_t(X_t, X_0)\|_p^\eta\right) . \tag{28}$$

- In the case of DLPM, our discussion leads us to the choice $p = 2$ and $\eta = 1$ (see (8)).
- In the case of LIM, the theory must rely on the choice $p = 2$ and $\eta = 2$ in order to obtain the denoising score matching loss equivalence (*i.e.*, L, L' are equivalent in (27)).

	Stochastic	Deterministic
Continuous (LIM)	$\frac{\overleftarrow{X}_t^\theta}{\gamma_t} - \frac{\alpha(1/\gamma_t - 1)}{\sigma_{1\to t}^{\alpha-1}}\hat{\epsilon}_t^\theta + (\frac{1}{\gamma_t^\alpha} - 1)^{1/\alpha}\epsilon'_t$	$\frac{\overleftarrow{X}_t^\theta}{\gamma_t} - \left(\frac{\sigma_{1\to t}^{1-\alpha}}{\gamma_t} - \sigma_{1\to t}^{1-\alpha}\right)\hat{\epsilon}_t^\theta$
Denoising (DLPM)	$\frac{\overleftarrow{Y}_t^\theta}{\gamma_t} - \Gamma_t\sigma_{1\to t}\hat{\epsilon}_t^\theta + \sqrt{\Gamma_t\Sigma_{1\to t-1}}G'_t$	$\frac{\overleftarrow{Y}_t^\theta}{\gamma_t} - \left(\frac{\sigma_{1\to t}}{\gamma_t} - \sigma_{1\to t-1}\right)\hat{\epsilon}_t^\theta$

Table 4: Distribution of $\overleftarrow{X}_{t-1}^\theta, \overleftarrow{Y}_{t-1}^\theta$. $\{G'_t\}_{t=T}^1$ are independent random variables distributed as $\mathcal{N}(0, \mathrm{I}_d)$, $\{\epsilon'_t\}_{t=T}^1$ are independent random variables distributed as $\mathcal{S}_\alpha^{\mathrm{i}}(0, \mathrm{I}_d)$. $\hat{\epsilon}_t^\theta$ is the model at hand at time t, the formula for $\Sigma_{1\to t}$ is given in (12), and $\Gamma_t = 1 - \gamma_t^2\Sigma_{1\to t-1}/\Sigma_{1\to t}$. Eventhough Γ_t involves two heavy-tailed random variables, it is bounded: $0 \leqslant \Gamma_t \leqslant 1$ (see Appendix C.3).

One must make the assumption that the losses L, L' are not infinite for some θ, which is not necessarily realistic because $S_t(X_t), S_t(X_t|X_0)$ are heavy-tailed random variables involving α-stable noise, and as such admit no variance.

- In the case of LIM, in the experiments the parameters $p = 1$ and $\eta = 1$ are chosen, instead of the previous squared loss, in order to obtain more stable training, potentially indicating that indeed L, L' (27) might be infinite.

F Technical Results

In this section, we give the proofs relative to our technique for faster training, as introduced in Appendix C.5.

Lemma 1. *Let $\bar{A}_0^t, \bar{A}_1^t$ bet two independent random variables distributed as $\mathcal{S}_{\alpha/2,1}(0, c_A)$. Define $Z_0 = Y_0$, and*

$$Z_t = \gamma_{1\to t}Z_0 + \sigma_{1\to t}\left(\bar{A}_1^t\right)^{1/2}G_t\ .$$

Moreover, let Z_{t-1} be equal to

$$Z_{t-1} = \frac{1}{\gamma_t}\left(Z_t - \Gamma'_t(\bar{A}_{0,1}^t)\sigma_{1\to t}\epsilon_t(Z_t, Z_0)\right) + \Sigma'_t(\bar{A}_{0,1}^t)G_{t-1}\ ,$$

where

$$\Sigma'_t(\bar{A}_{0,1}^t) = \Gamma'_t(\bar{A}_{0,1}^t)\sigma_{1\to t-1}\left(\bar{A}_0^t\right)^{1/2}$$
$$\Gamma'_t(\bar{A}_{0,1}^t) = \frac{\bar{A}_1^t\sigma_t^2}{\bar{A}_1^t\sigma_t^2 + \gamma_t^2\sigma_{1\to t-1}^2\bar{A}_0^t}$$
$$\epsilon_t(Z_t, Z_0) = \frac{Z_t - \gamma_{1\to t}Z_0}{\sigma_{1\to t}}\ .$$

Then the joint distribution of $(Z_0, Z_{t-1}, Z_t, \bar{A}_1^t, \bar{A}_0^t)$ matches the joint distribution of

$$\left(Y_0, Y_{t-1}, Y_t, A_t, \frac{\Sigma'_t(\bar{A}_{0,1}^t)}{\sigma_{1\to t}^2}\right)\ .$$

Proof. Consider the setting of Proposition 6. The distribution of Y_{t-1} given $Y_t, Y_0, A_{1:T}$ is characterized by the values of $\Sigma_{1\to t}, \Gamma_t$:

$$\tilde{\mathtt{m}}_{t-1} = \frac{1}{\gamma_t}\left(Y_t - \Gamma_t\sigma_{1\to t}\epsilon_t(Y_t, Y_0)\right), \quad \tilde{\Sigma}_{t-1}(A_{1:t}) = \Gamma_t(A_{1:t})\Sigma_{1\to t-1}(A_{1:t-1})\ ,$$

where

$$\Sigma_{1\to t}(A_{1:t}) = \sigma_t^2 A_t + \gamma_t^2\Sigma_{1\to t-1}(A_{1:t-1})$$
$$\epsilon_t(Y_t, Y_0) = \frac{Y_t - \gamma_{1\to t}Y_0}{\sigma_{1\to t}}$$
$$\Gamma_t(A_{1:t}) = 1 - \frac{\gamma_t^2\Sigma_{1\to t-1}(A_{1:t-1})}{\Sigma_{1\to t}(A_{1:t})}\ .$$

Directly applying the result of Lemma 2, we can affirm that

$$\Sigma_{1\to t-1}(A_{1:t-1}) \overset{d}{=} \sigma^2_{1\to t-1}\bar{A}^t_0 \ ,$$

where $\bar{A}^t_0 \sim \mathcal{S}_{\alpha/2,1}(0, c_A)$. In this conditions, the distribution of $\Gamma_t(A_{1:t})$ is equal to that of $\Gamma'_t(\bar{A}^t_{0,1})$, where

$$\Gamma'_t(\bar{A}^t_{0,1}) = 1 - \frac{\gamma_t^2\sigma^2_{1\to t-1}\bar{A}^t_0}{\sigma_t^2 A_t + \gamma_t^2\sigma^2_{1\to t-1}\bar{A}^t_0}$$

Since the distribution of Z_t does not change if we draw another independent $\bar{A}^t_1$ instead of A_t, this ends the proof. □

Lemma 2 (Sampling $\Sigma_{1\to t}$ with a single heavy-tailed r.v)**.** *Consider the setting of the data augmentation approach in Section 3.2.2, where in particular $\{A_t\}_{t=1}^T$ are independent random variables distributed according to $A_t \sim \mathcal{S}^1_{\alpha/2,1}(0, c_A)$, with $c_A = \cos^{2/\alpha}(\pi\alpha/4)$. Consider the random variable $\Sigma_{1\to t}(A_{1:t})$, as defined in (12):*

$$\Sigma_{1\to t}(A_{1:t}) = \sum_{k=1}^{t}\left(\frac{\gamma_{1\to t}}{\gamma_{1\to k}}\sqrt{A_k}\sigma_k\right)^2 \ .$$

Then

$$\Sigma_{1\to t}(A_{1:t}) \overset{d}{=} \sigma^2_{1\to t}A,$$

where $A \sim \mathcal{S}^1_{\alpha/2,1}(0, c_A)$.

Proof. By Proposition 5, Y_t given $Y_0, A_{1:t}$ is a random variable distributed as a Gaussian of variance $\Sigma_{1\to t}(A_{1:t})$:

$$Y_t \overset{d}{=} \gamma_{1\to t}Y_0 + \sqrt{\Sigma_{1\to t}(A_{1:t})}\bar{G}_t \ ,$$

where $\bar{G}_t$ is distributed as a standard Gaussian. Remember that Y_t and X_t admit the same distribution, with $X_t = \overset{d}{=} \gamma_{1\to t}X_0 + \bar{\epsilon}_t$ where $\bar{\epsilon}_t$ is distributed as a $\mathcal{S}^{\mathrm{i}}_\alpha(0, \mathrm{I}_d)$.

In the same spirit we can define a third sequence of random variables $\{Z_t\}_{t=0}^T$ with $Z_0 = X_0$, and

$$Z_t = \gamma_{1\to t}Z_0 + \sigma_{1\to t}\sqrt{A'_t}\bar{G}_t \ ,$$

where $\{A'_t\}_{t=0}^T$ are independent random variables distributed according to $A'_t \sim \mathcal{S}^1_{\alpha/2,1}(0, c_A)$. It is then quite clear from Section 2 that Z_t and Y_t admit the same distribution; in particular,

$$\sqrt{\Sigma_{1\to t}}G_t \overset{d}{=} \sigma_{1\to t}\sqrt{A'_t}G'_t \ .$$

From there, we use Lemma 3 to conclude that $\sqrt{\Sigma_{1\to t}} \overset{d}{=} \sigma_{1\to t}\sqrt{A'_t}$, which ends the proof. □

Lemma 3. *Let A, A' be positive real random variables, let Z be a real continuous random variable with density p_Z. Suppose that AZ and $A'Z$ admit the same distribution. Then A, A' admit the same distribution too.*

Proof. Let h be a measurable function. Then $\mathbb{E}(h(A)) = \mathbb{E}(h(AZ/Z)) = \mathbb{E}(h(A'Z/Z)) = \mathbb{E}(h(A'))$. This shows that A, A' have the same distribution. □

G Additional Experimental Details

All experiments are conducted using PyTorch. We use linear timesteps during training and sampling, and the scale-preserving process[6], being the only forward process readily provided by LIM. This entails choosing a sequence $\{\beta_t\}_{t=1}^T$ such that

$$\gamma_t = (1-\beta_t)^{1/\alpha}, \quad \sigma_t = (1-\gamma_t^\alpha)^{1/\alpha} \ ,$$

[6] we mention again that it is traditionally called the variance preserving process

resulting in $\sigma_{1\to t} = (1-\gamma_{1\to t}^{\alpha})^{1/\alpha}$ and $\gamma_{1\to t} = \prod_{i=1}^{t} \gamma_i$. With this choice, we obtain approximately $X_T \sim \mathcal{S}_\alpha(0, \mathrm{I}_d)$. We choose $\{\beta_i\}_{i=1}^{T}$ as the cosine schedule, as introduced by Nichol & Dhariwal (2021).

We do not give any of the heavy-tailed random variables $\{A_t\}_{t=1}^{T}$ as input to the neural network architecture, as we have witnessed worse performance in every scenarios we tried: learned embedding added to each model layer, concatenation to model input, concatenation at each layer, or feeding $\log(A_{1:T})$ instead of $A_{1:T}$ to better manage large jumps. This corresponds to the design choice D2 in Appendix C.4.2.

For image data generation with LIM, we use the same clipping hyper-parameters specified in Yoon et al. (2023).

All the training and experiments are conducted on four NVIDIA RTX8000 GPU and four NVIDIA V100 GPU, where a single training run on MNIST or CIFAR10_LT takes approximately 1 day per GPU, and requires about 4-12GB of VRAM for the batch sizes we use. Generating 5000 images with 1000 backward steps takes approximately 3-4 hours on one RTX8000 GPU.

G.1 2D Data

In the case of 2D data, we use 32000 datapoints for training, a batch size of 1024, and 25000 points for evaluation. Since we do not focus on the effect of diffusion steps, we set it to 100, where all methods have been observed to perform optimally.

The optimizer is Adam (Kingma & Ba (2017)) with learning rate 5e-3. We use a neural network consisting of four time-conditioned MLP blocks with skip connections, each of which consisting of two fully connected layers of width 64. The time t passes through two fully connected layers of size 32x32, and is fed to each time conditioned block, where it passes through an additional 32x64 fully connected layer before being component-wise added to the middle layer.

We use a mean squared logarithmic error (MSLE) loss designed to assess the fit to tails of distributions, and the precision/recall metric for our 2D datasets, as presented in Appendix G.3.

G.2 Image data

We work on the MNIST and the CIFAR10_LT dataset. CIFAR10_LT consists of the CIFAR10 images were artificial class unbalance has been introduced. The specific class counts we use are $[5000, 2997, 1796, 1077, 645, 387, 232, 139, 83, 50]$.

The optimizer is Adam (Kingma & Ba (2017)) with learning rate 1e-3 for MNIST and 2e-4 for CIFAR10_LT. We use the StepLR scheduler which scales the learning rate by $\gamma = .99$ every $N = 1000$ steps for CIFAR10_LT and $N = 400$ for MNIST.

To establish a fair comparison, LIM and DLPM use the same network model. We use a U-Net following the implementation of Nichol & Dhariwal (2021) available in `https://github.com/openai/improved-diffusion`. We dimension the network as follows: we set the hidden layers to $[128, 256, 256, 256]$, fix the number of residual blocks to 2 at each level, and add self-attention block at resolution 16x16, using 4 heads. We use an exponential moving average with a rate of 0.99 for MNIST and 0.9999 for CIFAR10_LT. We use the silu activation function at every layer. Diffusion time t is rescaled to $(0, 1)$ and fed to the model through the Transformer sinusoidal position embedding (Vaswani et al. (2023)). We train MNIST for 120000 steps with batch size 256 with a time horizon $T = 1000$, and CIFAR_LT for 400000 steps with batch size 100 with a time horizon $T = 4000$.

We use the FID metric for assessing the quality of our generative models, computing this metric between 5000 using images and 5000 generated images.

G.3 Metrics for generative models

MSLE we use a mean squared logarithmic error (MSLE) metric tailored to measure the fit on the tails of the distribution at hand. Drawing inspiration from Allouche et al. (2022), we define the MSLE as the squared distance between the logarithm of the inverse cumulative distributions of the

original and generated data. If $F, \hat{F}$ denote respectively the cumulative distribution function of the original data and the empirical cumulative distribution function of the generated data, then

$$\text{MSLE}(\xi) = \int_{\xi}^{1} \left(\log F^{-1}(p) - \log \hat{F}^{-1}(p) \right)^2 dp \ ,$$

where ξ is chosen the be 0.95.

Precision/recall These metrics are introduced in the setting of generative models by Sajjadi et al. (2018), and assess the overlap of sample distributions using local geometric structures. Precision measures how much the generated distribution is contained in the original data distribution (measuring quality), and recall measured how much of the original data distribution is covered by the generated distribution (diversity). We also consider the F_1^{pr} score which we define as the harmonic mean of these two values:

$$F_1^{\text{pr}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \ .$$

G.4 Additional results

We provide some more results on MNIST and CIFAR10_LT, with FID for non-isotropic noise, and with the F_1^{pr} metric for other methods (with clipping enabled in LIM and LIM-ODE). We also provide grid images in order to visually check the performance of DLPM.

MNIST	$\alpha = 1.5$	$\alpha = 1.7$	$\alpha = 1.8$	$\alpha = 1.9$	$\alpha = 2.0$
DLPM^{ni}	44.17	14.06	5.74	3.62	-
DLIM^{ni}	14.96	51.58	59.84	76.03	-

Table 5: FID↓, 1000 sampling steps for DLPM^{ni}, 25 sampling steps for DLIM^{ni}.

α	DLIM	DLPM	LIM	LIM-ODE	DDPM
1.7	0.884	**0.887**	0.857	0.869	-
1.8	0.874	**0.881**	0.821	0.875	-
1.9	0.877	**0.878**	0.700	0.808	-
2.0	0.820	0.871	0.694	0.772	**0.881**

Table 6: MNIST, $F_1^{\text{pr}} \uparrow$

α	DLIM	DLPM	LIM	LIM-ODE	DDPM
1.7	0.676	0.675	**0.679**	0.677	-
1.8	0.669	**0.680**	0.677	0.673	-
1.9	0.667	**0.669**	0.661	0.669	-
2.0	0.664	**0.667**	0.660	0.665	0.666

Table 7: CIFAR10_LT, $F_1^{\text{pr}} \uparrow$

Figure 6: MNIST, DLPM ($\alpha = 1.7$)

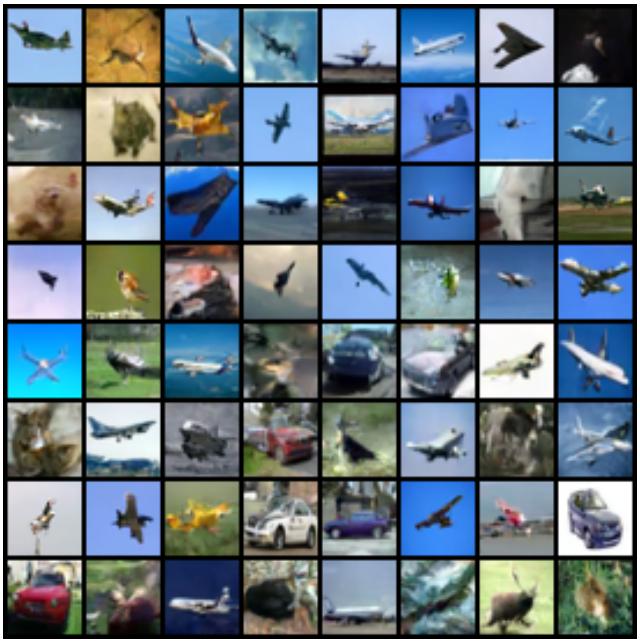


Figure 7: CIFAR10_LT, DLPM ($\alpha = 1.7$)