Denoising Lévy Probabilisitc Models - DLPM

Denoising Diffusion Models with Heavy Tails

Dario Shariatian, Umut Simsekli, Alain Durmus

September 17, 2024

Diffusion Process - discrete formulation (DDPM)

Advantages

- High quality samples
- Stable/easy training (e.g., contrary to GANs)
- Equivalence between multiple approaches

Disadvantages

- Lots of diffusion steps $n_s \gg 1$
- Mode collapse, especially with high class imbalance
- What if initial data distribution is heavy tailed (no variance)?

Proposal - change noising distribution

- Some previous work on other noise distributions exist
  - Generalized Gaussian distributions ([DSL21])
  - Gamma distributions ([NRW21])
- But show little success
  - No true time reversal
  - Hard sampling
  - Hard training
- We advocate for the α-**stable Lévy distributions**: generalize Gaussian with heavy tails.
- Lévy-Ito Models (LIM) have been proposed recently ([Yoo+23])
  - Continuous time formulation
  - But show limitations...

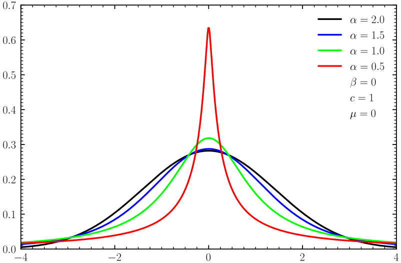
Proposal - alpha-stable heavy-tailed distribution

Explored solution: use heavy-tailed distributions for noising/denoising

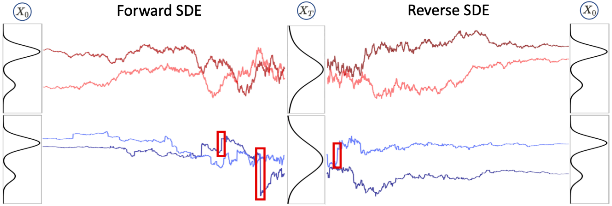
- Tackles the problem of generating a heavy-tailed data distribution.
- Less diffusion steps.
- Improvements on mode collapse and class imbalance.

Large jumps benefit the exploration of the data space?

α-stable Lévy distributions



(a) Symmetric α-Stable distribution, varying α [Wik24]



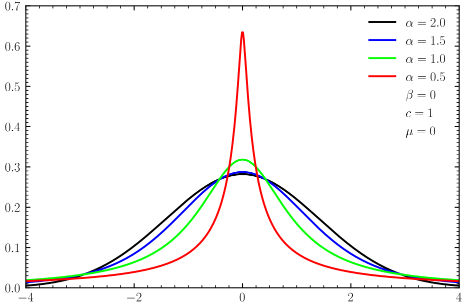
(b) Lévy Process vs Brownian Motion ($\alpha = 2$) [Yoo+23]

Definition and properties

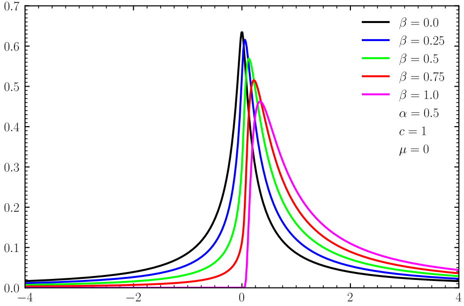
Notable special cases

- $(\beta = 0, \mu = 0)$: will be denoted $\mathcal{S}_\alpha(0, \sigma)$.
- $(\alpha = 2)$: $\mathcal{S}_\alpha(0, \sigma) = \mathcal{N}(0, 2\sigma^2)$.
- $(\alpha = 1)$: $\mathcal{S}_\alpha(0, 1)$ is the Cauchy distribution.

Definition and properties



(a) $\beta = 0, \mu = 0, \sigma = 1$, varying α [Wik24]



(b) $\alpha = 0.5, \mu = 0, \sigma = 1$, varying β [Wik24]

Gaussian Trick

Gaussian Trick

Gaussian Trick

Let $A \sim \mathcal{S}_{\alpha/2,1}(0, c_A)$, and $Z \sim \mathcal{N}(0, 1)$, where $c_A := \cos^{2/\alpha}(\pi\alpha/4)$. Then

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

- Defines many types of higher dimensional heavy tailed distributions.
- **Isotropic noise.** Draw a single $A \sim \mathcal{S}_{\alpha/2,1}(0, c_A)$, draw $Z \sim \mathcal{N}(0, \mathrm{I}_d)$, and compute

$$A^{1/2} Z \, . \tag{2}$$

- **Non-isotropic (independent) noise.** Draw a sequence $\{A_i\}_{i=1}^d$ i.i.d., draw $Z \sim \mathcal{N}(0, \mathrm{I}_d)$, and compute

$$A^{1/2} \odot Z \, . \tag{3}$$

Sampling an alpha-stable random variable

- CMS algorithm (J.M. Chambers, C.L. Mallows and B.W. Stuck).
- Generate $U \sim \mathcal{U}([-\pi/2, \pi/2])$, and $W \sim \mathcal{E}(1)$.
- ($\alpha \neq 1$) Compute:

$$X = (1+\zeta^2)^{1/2\alpha} \frac{\sin(\alpha(U+\xi))}{\cos(U)^{1/\alpha}} \left(\frac{\cos(U - \alpha(U+\xi))}{W} \right)^{(1-\alpha)/\alpha} \tag{4}$$

- ($\alpha = 1$) Compute:

$$X = \frac{1}{\xi} \left[\left(\frac{\pi}{2} + \beta U \right) \tan(U) - \beta \log \left(\frac{W \cos(u) \pi/2}{\zeta U + \pi/2} \right) \right] \tag{5}$$

- with

$$\zeta = -\beta \tan \frac{\pi\alpha}{2}\,, \qquad \xi = \begin{cases} \frac{1}{\alpha} \arctan(-\zeta) & \alpha \neq 1 \\ \frac{\pi}{2} & \alpha = 1 \end{cases} \tag{6}$$

- Then, $X \sim \mathcal{S}_{\alpha,\beta}(0,1)$. When $\alpha = 2, \beta = 0$, this is the Box-Muller algorithm.

Different multidimensional heavy-tailed distributions

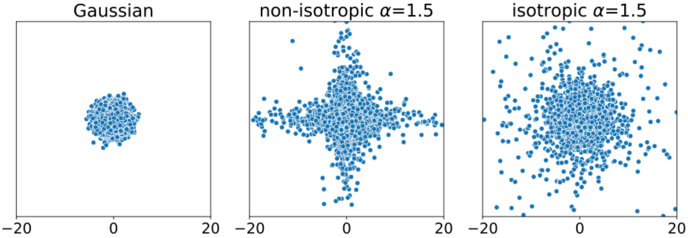


Figure: Different multidimensional heavy-tailed noise distributions, Gaussian vs $\alpha = 1.5$ [Yoo+23]

Forward Process - first approach

Forward Process - first approach

- The distribution of X_t given X_0 is given for any t by

$$X_t \overset{d}{=} \gamma_{1\to t} X_0 + \sigma_{1\to t} \bar{\epsilon}_t \tag{7}$$

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

$$\gamma_{1\to t} = \prod_{i=1}^{t} \gamma_t, \qquad \sigma_{1\to t} = \left(\sum_{i=1}^{t} \left(\frac{\gamma_{1\to t}}{\gamma_{1\to i}} \sigma_i \right)^{\alpha} \right)^{1/\alpha} . \tag{8}$$

Backward Process - first approach

- Consider $\{X_t\}_{t=0}^{n_s}$ the forward process defined earlier. We want to model and approximate the backward process similarly:

$$\overleftarrow{q}^{\theta}_{0:n_s}(x_{0:n_s}) = \overleftarrow{q}^{\theta}_{n_s}(x_{n_s}) \prod_{t=n_s}^{1} \overleftarrow{q}^{\theta}_{t-1|t}(x_{t-1}|x_t) , \tag{9}$$

such that $\overleftarrow{q}^{\theta}_{t-1|t} \approx p_{t-1|t}(x_{t-1}|x_t)$, with $p_{t-1|t}$ the density of the distribution of X_{t-1} given X_t.

- $p_{t|t-1}(x_t|x_{t-1}), p_{t|0}(x_t|x_0)$ have analytical expressions. No known techniques to characterize

$$p_{t-1|t}(x_{t-1}|x_t) , \quad p_{t-1|t,0}(x_{t-1}|x_t, x_0) \tag{10}$$

- **How to design the approximation for the backward process?**
- Our approach: using data augmentation and the "Gaussian trick"

Forward process - data augmentation approach

Forward process - data augmentation approach

- The distribution of Y_t given $Y_0, \{A_t\}_{t=1}^{n_\mathrm{s}}$ is characterized by the following:

$$Y_t \stackrel{d}{=} \gamma_{1\to t} Y_0 + \Sigma_{1\to t}(A_{1:t})^{1/2} \bar{G}_t, \tag{11}$$

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

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

Forward process - data augmentation approach

Backward process - data augmentation approach

Backward process - data augmentation approach

- Let's consider $\{Y_t\}_{t=0}^{n_s}$, and condition on $\{A_t\}_{t=1}^{n_s}$. Then:

$$p_{t-1|t,0,a}(y_{t-1}|y_t, y_0, a_{1:n_s}) = \phi_d(y_{t-1}; \tilde{\mathtt{m}}_{t-1}(y_t, y_0, a_{1:t}), \tilde{\Sigma}_{t-1}(a_{1:t})) \;, \tag{13}$$

where ϕ_d is the density of the standard Gaussian, and

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

with

$$\epsilon_t(y_t, y_0) = \frac{y_t - \gamma_{1\to t} y_0}{\sigma_{1\to t}} \;, \quad \Sigma_{1\to t}(a_{1:t}) = \sum_{k=1}^{t}\left(\frac{\gamma_{1\to t}}{\gamma_{1\to k}}\sqrt{a_k}\sigma_k\right)^2 \;, \quad \Gamma_t(a_{1:t}) = 1 - \frac{\gamma_t^2 \Sigma_{1\to t-1}(a_{1:t-1})}{\Sigma_{1\to t}(a_{1:t})} \;. \tag{15}$$

Note that Γ_t is bounded: $0 \leqslant \Gamma_t \leqslant 1$.

Backward process - model

Reminder: Loss function - Gaussian case

- Consider the KL loss $\mathscr{L}^{\mathrm{D}} : \theta \mapsto \mathrm{KL}(p_\star \| \overleftarrow{q}^{\theta}_{0})$:

$$\mathscr{L}^{\mathrm{D}}(\theta) \leqslant \mathscr{L}^{\mathrm{D}}_{n_{\mathrm{s}}} + \sum_{t=2}^{n_{\mathrm{s}}} \mathscr{L}^{\mathrm{D}}_{t-1}(\theta) + \mathscr{L}^{\mathrm{D}}_{0}(\theta) + C \tag{16}$$

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

$$\mathscr{L}^{\mathrm{D}}_{n_{\mathrm{s}}} = \mathbb{E}\left[\mathrm{KL}\left(p_{t|0}(\cdot|X_0) \| \mathcal{N}(0, \sigma_{1\to t}\mathrm{I}_d)\right)\right] \tag{17}$$

$$\mathscr{L}^{\mathrm{D}}_{0}(\theta) = -\mathbb{E}\left[\log\left(\overleftarrow{q}^{\theta}_{0|1}(X_0|X_1)\right)\right] \tag{18}$$

$$\mathscr{L}^{\mathrm{D}}_{t-1}(\theta) = \mathbb{E}\left[\mathrm{KL}\left(p_{t-1|0,t}(\cdot|X_0, X_t) | \overleftarrow{q}^{\theta}_{t-1|t}(\cdot|X_t)\right)\right] . \tag{19}$$

- For a fixed variance $\hat{\Sigma}^{\theta}_{t-1} = \tilde{\Sigma}_{t-1}$, with $\tilde{\Sigma}_{t-1}$ given in (14), one resorts to optimize a convenient loss function:

$$\mathscr{L}^{\mathrm{D}}_{t-1}(\theta) = \lambda_t \|\tilde{\mathtt{m}}_{t-1}(x_t, x_0) - \hat{\mathtt{m}}^{\theta}_{t-1}(x_t)\|^2, \tag{20}$$

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

Loss function - alpha-stable case

- **A naive solution:** by Jensen's inequality:

$$\mathrm{KL}(p_\star \| \overleftarrow{q}_0^{\theta}) \leqslant \mathbb{E}\left(\mathrm{KL}\left[p_\star(\cdot) \| \overleftarrow{q}_{0|a}^{\theta}(\cdot | A_{1:n_s})\right]\right) \;. \tag{21}$$

- As we see in (20), this expression would involve taking expectation of A_t;

- However, it is distributed as $\mathcal{S}_{\alpha/2,1}(0, c_A)$ and admits no first order moment.

Loss function - alpha-stable case

- We consider the following loss function:

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

$$\mathscr{L}^{\mathrm{L}}_{t-1}(\theta, A_{1:n_{\mathrm{s}}}) := \mathbb{E}\left[\mathrm{KL}\left(p_{t-1|t,0,a}(\cdot|Y_t, Y_0, A_{1:n_{\mathrm{s}}}) \parallel \overleftarrow{q}^{\theta}_{t-1|t,a}(\cdot|Y_t, A_{1:n_{\mathrm{s}}})\right) \middle| A_{1:n_{\mathrm{s}}}\right] , \tag{23}$$

and $p_{t-1|0,t,a}$ denotes the conditional density of Y_{t-1} given Y_0, Y_t and $A_{1:n_{\mathrm{s}}}$.

- Since $p_{t-1|t,0,a}$ and $\overleftarrow{q}^{\theta}_{t-1|t,a}$ are Gaussian (thanks to the conditioning), the KL term has a closed-form formula, as in the case of DDPM.

Loss function - design choice **D1**

- Recall we considered the following model:

$$\overleftarrow{q}^{\,\theta}_{t-1|t}(x_{t-1}|x_t) = \int \overleftarrow{q}^{\,\theta}_{t-1|t,a}(x_{t-1}|x_t, a_{1:n_\mathrm{s}})\ \psi^{(\alpha)}_{1:n_\mathrm{s}}(a_{1:n_\mathrm{s}})\mathrm{d}a_{1:n_\mathrm{s}} \tag{24}$$

  with

$$\overleftarrow{q}^{\,\theta}_{t-1|t,a}(x_{t-1}|x_t, a_{1:n_\mathrm{s}}) = \phi_d(y_{t-1} \mid \hat{\mathtt{m}}^{\theta}_{t-1}(y_t, a_{1:n_\mathrm{s}}), \hat{\Sigma}^{\theta}_{t-1}(a_{1:n_\mathrm{s}}))\ , \tag{25}$$

  where ϕ_d is the density of the d-dimensional Gaussian distribution.

- **D1.** We set a fixed variance $\hat{\Sigma}^{\theta}_{t}(a_{1:t}) = \tilde{\Sigma}_t(a_{1:t})$

- Recall:

$$p_{t-1|t,0,a}(y_{t-1}|y_t, y_0, a_{1:n_\mathrm{s}}) = \phi_d(y_{t-1}; \tilde{\mathtt{m}}_{t-1}(y_t, y_0, a_{1:t}), \tilde{\Sigma}_{t-1}(a_{1:t}))\ , \tag{26}$$

Loss function - design choice **D2**

- **D2.** Since

$$\tilde{\mathtt{m}}_{t-1}(Y_t, Y_0, A_{1:n_{\mathrm{s}}}) = \frac{1}{\gamma_t}\left(Y_t - \sigma_{1\to t}\Gamma_t(A_{1:n_{\mathrm{s}}})\epsilon_t(Y_t, Y_0)\right), \tag{27}$$

  we parameterize $\hat{\mathtt{m}}^{\theta}_{t-1}$ using $\hat{\epsilon}^{\theta}_t$:

$$\hat{\mathtt{m}}^{\theta}_{t-1}(Y_t, A_{1:t}) = \frac{1}{\gamma_t}\left(Y_t - \sigma_{1\to t}\Gamma_t(A_{1:t})\hat{\epsilon}^{\theta}_t(Y_t, A_{1:t})\right). \tag{28}$$

- Then, $\mathscr{L}^{\mathrm{L}}_{t-1}$ becomes

$$\mathscr{L}^{\mathrm{L}}_{t-1}(\theta) = \mathbb{E}\left[\lambda^2_{t,\Gamma_t}\|\hat{\epsilon}^{\theta}_t(Y_t, A_{1:n_{\mathrm{s}}}) - \epsilon_t(Y_t, Y_0)\|^2\right], \tag{29}$$

  where

$$\lambda_{t,\Gamma_t} = \frac{\Gamma_t\sigma_{1\to t}}{2\gamma_t\tilde{\Sigma}_{t-1}} \quad \text{and} \quad \epsilon_t(Y_t, Y_0) = \frac{Y_t - \gamma_{1\to t}Y_0}{\sigma_{1\to t}}. \tag{30}$$

- We will stick to the common choice of choosing $\lambda = 1$ [Yan+24].
- Other choices and optimizations are left to further work.

Loss function - design choice **D3**

- **D3.** We drop the dependency of $\hat{\epsilon}_t^\theta$ on $\{A_t\}_{t=1}^{n_s}$. Thus $\hat{\epsilon}_t^\theta$ only depends on t, Y_t

- Better performance in our experiments, allows further tricks, and enables one to re-use existing neural network architectures.

Simplified loss function - alpha-stable case

- With the design choices **D1, D2, D3**, we obtain the simplified denoising objective function:

$$\boxed{\mathscr{L}^{\mathrm{Simple}}_{t-1}(\theta) = \mathbb{E}\left[\mathbb{E}\left(\|\hat{\epsilon}^{\theta}_t(Y_t) - \epsilon_t(Y_t, Y_0)\|^2 \;\middle|\; A_{1:n_{\mathrm{s}}}\right)^{1/2}\right], \qquad t \in \{2, \cdots, n_{\mathrm{s}}\}} \tag{31}$$

with $G_t \sim \mathcal{N}(0, \mathrm{I}_d)$, $A_t \sim \mathcal{S}_{\alpha/2,1}(0, c_A)$,

$$Y_t = \gamma_{1\to t} Y_0 + \Sigma_{1\to t}(A_{1:t})^{1/2} G_t\,, \qquad \epsilon_t(Y_t, Y_0) = \frac{Y_t - \gamma_{1\to t} Y_0}{\sigma_{1\to t}}\,, \tag{32}$$

$$\hat{\mathtt{m}}^{\theta}_{t-1}(Y_t, A_{1:t}) = \frac{1}{\gamma_t}\left(Y_t - \sigma_{1\to t}\Gamma_t(A_{1:t})\hat{\epsilon}^{\theta}_t(Y_t)\right), \qquad \hat{\Sigma}^{\theta}_{t-1}(A_{1:t}) = \Gamma_t(A_{1:t})\Sigma_{1\to t-1}(A_{1:t-1})\,, \tag{33}$$

where

$$\Sigma_{1\to t-1}(A_{1:t-1}) = \sum_{k=1}^{t-1}\left(\frac{\gamma_{1\to t-1}}{\gamma_{1\to k}}\sqrt{A_k}\sigma_k\right)^2, \qquad \Sigma_{1\to t}(A_{1:t}) = \sigma_t^2 A_t + \gamma_t^2 \Sigma_{1\to t-1}(A_{1:t-1})\,, \tag{34}$$

and $\Gamma_t = 1 - \dfrac{\gamma_t^2 \Sigma_{1\to t-1}(A_{1:t-1})}{\Sigma_{1\to t}(A_{1:t})}$.

Bonus - faster sampling

- Assume the design choices **D1, D2, D3** are satisfied. Then one can obtain the following simplified denoising objective function:

$$\boxed{\mathscr{L}_{t-1}^{\mathrm{SimpleLess}}(\theta) = \mathbb{E}\left[\mathbb{E}\left(\|\hat{\epsilon}_t^{\theta}(Z_t) - \epsilon_t(Z_t, Z_0)\|^2 \mid \bar{A}_{t-1}, \bar{A}_t\right)\right]^{1/2}, \qquad t \in \{2, \cdots, n_{\mathrm{s}}\}} \tag{35}$$

with $G_t \sim \mathcal{N}(0, \mathrm{I}_d)$, $\bar{A}_t, \bar{A}_{t-1} \sim \mathcal{S}_{\alpha/2,1}(0, c_A)$,

$$Z_t = \gamma_{1\to t} Z_0 + \Sigma_t'^{1/2}(\bar{A}_{t,t-1}) G_t\,, \qquad \epsilon_t(Z_t, Z_0) = \frac{Z_t - \gamma_{1\to t} Z_0}{\sigma_{1\to t}}\,, \tag{36}$$

$$\hat{\mathtt{m}}_{t-1}^{\theta}(Z_t, \bar{A}_{t,t-1}) = \frac{1}{\gamma_t}\left(Z_t - \sigma_{1\to t}\Gamma_t'(\bar{A}_{t,t-1})\hat{\epsilon}_t^{\theta}(Z_t)\right), \quad \hat{\Sigma}_{t-1}^{\theta}(\bar{A}_{t,t-1}) = \Gamma_t'(\bar{A}_{t,t-1})\Sigma_{t-1}'(\bar{A}_{t-1})\,, \tag{37}$$

where

$$\Sigma_{t-1}'(\bar{A}_{t-1}) = \sigma_{1\to t-1}^2 \bar{A}_{t-1}, \quad \Sigma_t'(\bar{A}_{t,t-1}) = \sigma_t^2 \bar{A}_t + \gamma_t^2 \Sigma_{t-1}'(\bar{A}_{t-1})\,, \tag{38}$$

and $\Gamma_t'(\bar{A}_t, \bar{A}_{t-1}) = 1 - \dfrac{\gamma_t^2 \Sigma_{t-1}'(\bar{A}_{t-1})}{\Sigma_t'(\bar{A}_{t,t-1})}$.

Bonus - faster sampling

- Assume the design choices **D1, D2, D3** are satisfied. Then one can obtain the following simplified denoising objective function:

$$\mathscr{L}_{t-1}^{\text{SimpleLess}}(\theta) = \mathbb{E}\left[\mathbb{E}\left(\|\hat{\epsilon}_t^{\theta}(Z_t) - \epsilon_t(Z_t, Z_0)\|^2 \mid \bar{A}_{t-1}, \bar{A}_t\right)\right]^{1/2}, \qquad t \in \{2, \cdots, n_{\mathrm{s}}\} \tag{39}$$

- Essentially $\bar{A}_t \stackrel{d}{=} A_t$ and $\sigma^2_{1\to t-1}\bar{A}_{t-1} \stackrel{d}{=} \Sigma_{1\to t-1}(A_{1:t-1})$.

- Much cheaper than sampling Y_t given Y_0 (must sample $A_{1:t}$ for each datapoint).

DLIM - Denoising Lévy Implicit Models

- We obtain a deterministic sampling process, with the same techniques as in DDIM ([SME20]).
- The process $\{Z_t\}_{t=0}^{n_s}$ is such that:

$$Z_0 \sim p_\star \ , \quad Z_{n_s} \sim \mathcal{S}_\alpha\left(\gamma_{1\to n_s} Z_0, \sigma_{1\to n_s} \mathrm{I}_d\right) \ , \quad \text{and} \tag{40}$$

$$Z_{t-1} = \gamma_{1\to t-1} Z_0 + (\sigma_{1\to t-1}^\alpha - \rho_t^\alpha)^{1/\alpha} \epsilon_t(Z_t, Z_0) + \rho_t A_t^{1/2} G_t \ , \tag{41}$$

with $\{G_t\}_{t=1}^{n_s}$ i.i.d. $\mathcal{N}(0, \mathrm{I}_d)$, $\{A_t\}_{t=1}^{n_s}$ i.i.d. $\mathcal{S}_{\alpha/2,1}(0, c_A)$, and $\{\rho_t\}_{t=1}^{n_s}$ an alternative noise schedule.

- Designed such that $Z_t | Z_0 \overset{d}{=} Y_t | Y_0$ for $t \in \{1, \cdots, n_s\}$.
- One can use the same model $\hat{\epsilon}_t^\theta(Z_t) \approx \epsilon_t(Z_t, Z_0)$ trained for DLPM.

LIM vs DLPM

`https://openreview.net/forum?id=0Wp3VHX0Gm`

- LIM is the continuous time competition: extending the SDE formulation to Levy processes.
- DLPM leverages the flexibility of the discrete formulation for diffusion.
- Much simpler and accessible theory.
- Different training loss, different sampling algorithms for the backward process.

LIM - forward

- The forward process X_t, with $X_0 \sim p_\star$, is written

$$\mathrm{d}X_t = \gamma(t, X_{t-})\mathrm{d}t + \sigma(t)\mathrm{d}L_t^\alpha, \tag{42}$$

where X_{t-} denotes the left limit of X at time t. LIM only defines scale-preserving schedule:

$$\gamma(t, x) = -\frac{\beta_t}{\alpha}x, \quad \sigma(t) = \beta_t^{1/\alpha}. \tag{43}$$

- Similarly, one can explicitly characterize the distribution of X_t given X_0:

$$X_t \overset{d}{=} \gamma_{1\to t}X_0 + \sigma_{1\to t}\bar{\epsilon}, \tag{44}$$

where $\bar{\epsilon}_t \sim \mathcal{S}_\alpha^{\mathrm{i}}(0, \mathrm{I}_d)$. The values of the continuous $\gamma_{1\to t}$ and $\sigma_{1\to t}$ match with their previous definition on integer timesteps.

LIM - backward

- We consider the following backward process $\overleftarrow{X}_t$:

$$\mathrm{d}\overleftarrow{X}_t = \left(-\gamma(t, \overleftarrow{X}_{t+}) + \alpha\sigma^{\alpha}(t, \overleftarrow{X}_{t+}) S_t^{(\alpha)}(\overleftarrow{X}_{t+})\right) dt + \sigma(t)\mathrm{d}\bar{L^{\alpha}}_t + d\bar{Z}_t \tag{45}$$

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
- $S_t^{(\alpha)}$ is the fractional score function:

$$S_t^{(\alpha)}(x) = \frac{\Delta^{\frac{\alpha-2}{2}} \nabla p_t(x)}{p_t(x)} , \tag{46}$$

where $\Delta^{\eta/2}$ is the fractional Laplacian of order $\eta/2$, defined with Fourier transform $\mathcal{F}$:

$$\Delta^{\eta/2} f(x) = \mathcal{F}^{-1}\{\|u\|^{\eta}\mathcal{F}\{f\}(u)\} . \tag{47}$$

LIM - training

- The true score $S_t^{(\alpha)}(x_t|x_0)$ can be expressed as

$$S_t^{(\alpha)}(x_t|x_0) = -\frac{1}{\alpha\sigma_{1\to t}^{\alpha-1}(t)}\epsilon_t(x_t, x_0), \tag{48}$$

where $\epsilon_t(x_t, x_0) = \frac{x_t - \gamma_{1\to t}x_0}{\sigma_{1\to t}}$, thus we re-parametrize

$$s_\theta(x_t, t) = -\frac{1}{\alpha\sigma_{1\to t}^{\alpha-1}(t)}\hat{\epsilon}_t^\theta(x_t, x_0), \tag{49}$$

so that we rather work with $\hat{\epsilon}_t^\theta$.

- Training loss obtained using denoising score matching technique:

$$L : \theta \mapsto \mathbb{E}\|s_\theta(X_t, t) - S_t^{(\alpha)}(X_t)\|^2, \qquad L' : \theta \mapsto \mathbb{E}\|s_\theta(X_t, t) - S_t^{(\alpha)}(X_t|X_0)\|^2, \tag{50}$$

are equivalent objective functions, with s_θ the score approximation given by the model.

LIM vs DLPM - forward/backward

With $\{G'_t\}_{t=n_s}^{1}$ i.i.d. $\mathcal{N}(0, \mathrm{I}_d)$, $\{\epsilon'_t\}_{t=n_s}^{1}$ i.i.d. $\mathcal{S}_\alpha^{\mathrm{i}}(0, \mathrm{I}_d)$, and $\hat{\epsilon}_t^\theta$ the model at time t:

	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)	$\frac{\overleftarrow{Y}_t^\theta}{\gamma_t} - \Gamma_t \sigma_{1\to t} \hat{\epsilon}_t^\theta + \Gamma_t \Sigma_{1\to t-1} G'_t$	$\frac{\overleftarrow{Y}_t^\theta}{\gamma_t} - \left(\frac{\sigma_{1\to t}}{\gamma_t} - \sigma_{1\to t-1} \right) \hat{\epsilon}_t^\theta$

- **Stochastic sampling** Different sampling procedures. Moreover:
  1. When $\alpha = 2$, $0 \leqslant \Gamma_t \leqslant 1$ becomes deterministic, and one recovers DDPM formulas
  2. Γ_t brings additional stochasticity
  3. Γ_t scales (i) the noise added at time $t-1$ (ii) the output of the noise model.
- **Deterministic sampling** Different sampling procedures.

LIM vs DLPM - training

- Alike the Gaussian case ($\alpha = 2$), the score $S_t^{(\alpha)}(x_t|x_0)$ is a linear expression of the noise term:

$$S_t^{(\alpha)}(x_t|x_0) = -\frac{1}{\alpha\sigma_{1\to t}^{\alpha-1}(t)}\epsilon_t(x_t, x_0) \ , \tag{51}$$

  leading to a similar denoising loss:

$$\mathcal{L}_{t-1} : \theta \mapsto \mathbb{E}\left(\|\hat{\epsilon}_t^{\theta}(X_t) - \epsilon_t(X_t, X_0)\|_p^{\eta}\right) . \tag{52}$$

- DLPM: use $p = 2$ and $\eta = 1$.
- LIM (theory): use $p = 2$ and $\eta = 2$, for denoising score matching loss equivalence. But $\epsilon_t(X_t, X_0)$ is heavy-tailed: no variance!
- LIM (experiments): use $p = 1$ and $\eta = 1$. Indicates potential shortcoming of the theoretical approach.

Setup

- Our loss function

$$\mathscr{L}^{\mathrm{Simple}}(\theta) = \sum_{t=1}^{n_{\mathrm{s}}} \mathbb{E}\left[\mathbb{E}\left(\|\hat{\epsilon}_t^{\theta}(Y_t) - \epsilon_t(Y_t, Y_0)\|^2 \mid A_{1:n_{\mathrm{s}}}\right)^{1/2}\right] \tag{53}$$

involves an expectation with respect to $A_{1:n_{\mathrm{s}}}$. We propose the *median-of-means* estimator ([LM19]), denoted by DLPM_5 ($M = 5$).

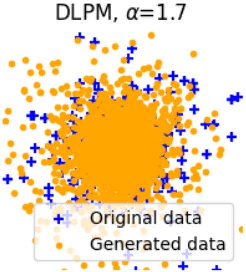
- We experiment with non-isotropic diffusion $\mathrm{DLPM}^{\mathrm{ni}}$.

- We consider the range $1.5 \leqslant \alpha \leqslant 2.0$, otherwise training/sampling get unstable.

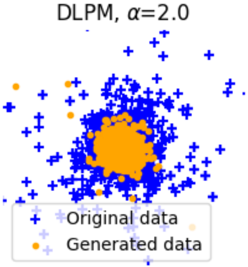
- We use the CIFAR10_LT (long tail), unbalanced modification of the CIFAR10 ([Yoo+23]).
  - Class count: $[5000, 2997, 1796, 1077, 645, 387, 232, 139, 83, 50]$.

2D data - covering the dataset and capturing heavy-tails

- **Dataset** 20000 samples of $\mathcal{S}^{\mathrm{i}}_{\alpha}(0, 0.05 \cdot \mathrm{I}_2)$, with $\alpha = 1.7$.
- Main challenge: cover the dataset and correctly capture the tails.



(a) DLPM with $\alpha = 1.7$



(b) DLPM with $\alpha = 2.0$ (DDPM)

- The lighter tailed process fails to capture the distribution's tail.

2D data - covering the dataset and capturing heavy-tails

- Drawing inspiration from [AGG22], we define the MSLE:

$$\mathrm{MSLE}(\xi) = \int_{\xi}^{1} \left(\log F^{-1}(p) - \log \hat{F}^{-1}(p)\right)^2 dp \,, \tag{54}$$

where $F, \hat{F}$ denote respectively the cdf of the true data and the generated data.

Method	1.7	1.8	1.9	2.0
DLPM	**0.071** $\pm$ 0.028	**0.099** $\pm$ 0.044	**0.132** $\pm$ 0.101	**0.798** $\pm$ 0.601
LIM	0.267 $\pm$ 0.077	0.653 $\pm$ 0.413	2.444 $\pm$ 1.067	1.239 $\pm$ 0.240

Table: $\mathrm{MSLE}_{\xi=0.95} \downarrow$ averaged over 20 runs

2D data - managing class imbalance

- **Dataset** Mixture of nine Gaussian distributions arranged in a grid

$$\sum_{i=1}^{9} w_i \mathcal{N}(\mu_i, 0.05^2 \cdot \mathrm{I}_2)\ . \tag{55}$$

Mixture weights range from .01 to .3: $\{.01, .02, .02, .05, .05, .1, .1, .15, .2, .3\}$.

- Main challenge: correctly guess the mixture weights

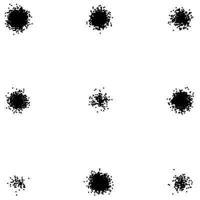


Figure: Gaussian grid

Method	$\alpha = 1.7$	$\alpha = 1.8$	$\alpha = 1.9$	$\alpha = 2.0$
DLPM	0.78 ± 0.04	0.75 ± 0.05	0.75 ± 0.04	$\mathbf{0.71} \pm 0.03$
DLPM_5	$\mathbf{0.79} \pm 0.03$	$\mathbf{0.77} \pm 0.08$	$\mathbf{0.80} \pm 0.05$	0.69 ± 0.05
$\mathrm{DLPM}^{\mathrm{ni}}$	0.71 ± 0.02	0.77 ± 0.05	0.77 ± 0.05	0.70 ± 0.04
LIM	0.72 ± 0.02	0.63 ± 0.05	0.62 ± 0.02	0.65 ± 0.02

Table: $F_1^{\mathrm{pr}} = 2\frac{\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}} \uparrow$ score, averaged over 30 runs

2D data - faster convergence

- DLIM vs LIM-ODE with varying total diffusion steps n_s, on the Gaussian grid.
- Main challenge: get to the data distribution with the smallest n_s possible

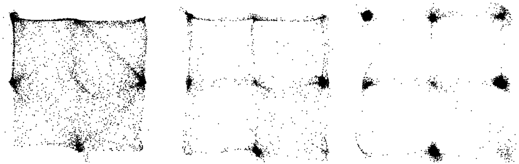


Figure: DLIM with $n_s = 5, 10, 25$ diffusion steps on the Gaussian grid

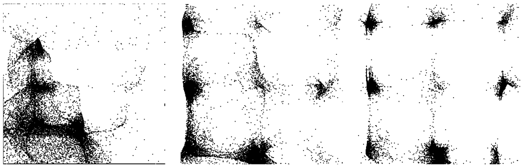


Figure: LIM-ODE with $n_s = 5, 10, 25$ diffusion steps on the Gaussian grid

Image data - LIM vs DLPM

- **Dataset** MNIST and CIFAR10_LT.
- Convergence speed for the different methods, varying total number of diffusion steps n_{s}.

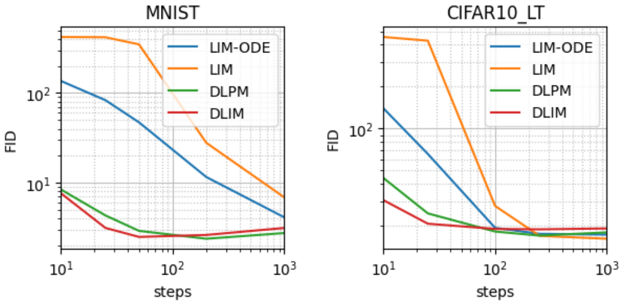


Figure: FID↓ with varying step size, $\alpha = 1.7$

Image data - LIM vs DLPM

MNIST	$\alpha = 1.5$	$\alpha = 1.7$	$\alpha = 1.8$	$\alpha = 1.9$	$\alpha = 2.0$
LIM	4.075	5.171	6.812	11.202	11.693
DLPM^{ni}	44.173	14.055	5.739	3.618	-
DLPM_5	**3.801**	3.030	**2.506**	**2.705**	-
DLPM	5.392	**2.938**	2.930	3.237	3.632
LIM-ODE	45.717	68.153	85.090	113.196	29.04
DLIM_5	14.959	51.582	59.841	76.033	-
DLIM_5	**3.373**	2.931	3.440	4.314	-
DLIM	3.376	**2.811**	**3.178**	**3.273**	5.183
CIFAR10_LT					
LIM	16.13	**16.21**	**17.67**	**19.24**	21.56
DLPM	**16.10**	18.00	19.94	20.21	21.07
LIM-ODE	30.170	65.788	84.559	101.704	32.00
DLIM	**20.699**	**20.775**	**21.967**	**22.799**	23.999

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

- Better performance of DLPM as compared to LIM.
- Better performance with smaller α.

Reference

[LM19] Gábor Lugosi and Shahar Mendelson. "Mean estimation and regression under heavy-tailed distributions: A survey". In: *Foundations of Computational Mathematics* 19.5 (2019), pp. 1145–1190.

[SME20] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In: *CoRR* abs/2010.02502 (2020). arXiv: 2010.02502. URL: https://arxiv.org/abs/2010.02502.

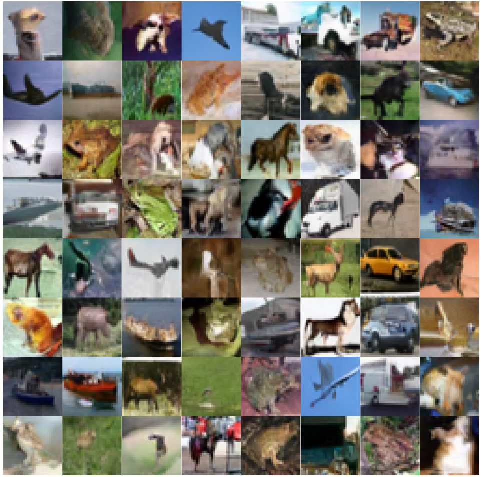
[DSL21] Jacob Deasy, Nikola Simidjievski, and Pietro Lio'. "Heavy-tailed denoising score matching". In: *ArXiv* abs/2112.09788 (2021). URL: https://api.semanticscholar.org/CorpusID:245334465.

[NRW21] Eliya Nachmani, Robin San Roman, and Lior Wolf. *Denoising Diffusion Gamma Models*. 2021. arXiv: 2110.05948 [eess.SP].

[AGG22] Michaël Allouche, Stéphane Girard, and Emmanuel Gobet. "EV-GAN: Simulation of extreme events with ReLU neural networks". In: *Journal of Machine Learning Research* 23.150 (2022), pp. 1–39. URL: https://hal.science/hal-03250663.

[Yoo+23] Eun Bi Yoon et al. "Score-based Generative Models with Lévy Processes". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., 2023, pp. 40694–40707. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/8011b23e1dc3f57e1b6211ccad498919-Paper-Conference.pdf.

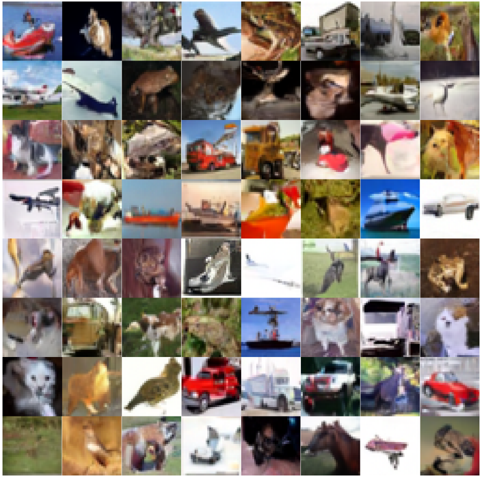
Some images - DLPM



(a) CIFAR10, $n_s = 4000$

(b) MNIST, $n_s = 1000$

Some images - DLIM



(a) CIFAR10, $n_s = 200$

(b) MNIST, $n_s = 50$