Non-asymptotic Analysis of Diffusion Annealed Langevin Monte Carlo for Generative Modelling

Paula Cordero Encinar, Deniz Akyildiz and Andrew Duncan

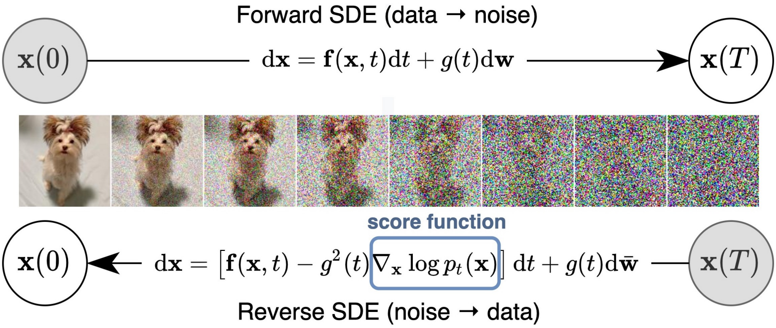
Imperial College London

2nd RSS/Turing Workshop on Gradient Flows for Sampling, Inference, and Learning **24/03/2025**

Introduction: Generative Models

The **goal** of generative modelling is to learn the underlying probability distribution π_{data} given a set of samples.

In particular, diffusion models achieve this as follows:



Introduction: Diffusion Models

- The forward process in diffusion models is typically an Ornstein-Uhlenbeck process:

$$\mathrm{d}X_t = -X_t \mathrm{d}t + \sqrt{2}\mathrm{d}B_t, \quad \text{for } 0 \leq t \leq T.$$

where $(B_t)_{t \in [0,T]}$ is a Brownian motion on $\mathbb{R}^d$ and $X_0 \sim \pi_{\text{data}}$.

! Disclaimer: The OU process takes ∞ time to interpolate between π_{data} and a Gaussian.

Introduction: Diffusion Models

- At generation time, these models evolve samples along a path of probability distributions $(\mu_t)_{t\in[0,T]}$. The intermediate random variables $X_t \sim \mu_t$ are defined as

$$X_t = \sqrt{\lambda_t} X + \sqrt{1-\lambda_t} Z,$$

  for $t \in [0, T]$, where $X \sim \pi_{\text{data}}$, $Z \sim \mathcal{N}(0, I)$ is independent of X and a schedule $\lambda_t = \min\{1, e^{-2(T-t)}\}$.

Remark: μ_t is given by a convolution.

Note: We reverse the notation wrt diffusion models: $\mu_T = \pi_{\text{data}}$ (ours) vs $\mu_0 = \pi_{\text{data}}$

Introduction: Diffusion vs Geometric Path

Motivation: Let $\pi_{\text{data}} = (1 - e^{-m^2/4})\mathcal{N}(m, 1) + e^{-m^2/4}u_m$, where u_m is the smoothed uniform distribution on $I_m = [-m, 2m]$ for $m = 10$ (Chehab et al. (2024)) and $\nu = \mathcal{N}(0, 1)$.

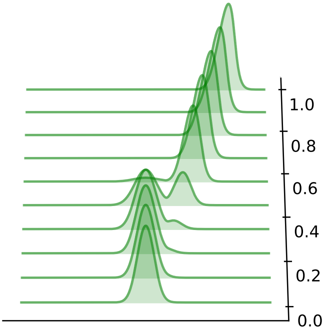


Figure 1: Geometric path

$$\mu_t(x) = \pi_{\text{data}}{}^{\lambda_t}(x)\nu^{1-\lambda_t}(x)$$

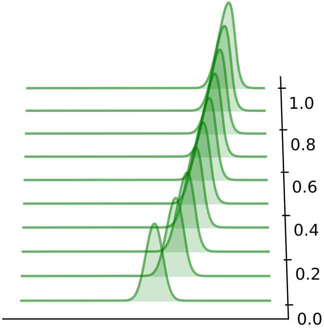


Figure 2: Gaussian Diffusion path

$$\mu_t(x) = \frac{\pi_{\text{data}}(x/\sqrt{\lambda_t})}{\lambda_t{}^{d/2}} * \frac{\nu\left(x/\sqrt{1-\lambda_t}\right)}{(1-\lambda_t)^{d/2}}$$

Introduction: Diffusion vs Geometric Path

What was the previous figure trying to show?

Proposition

If π_{data} has a finite log-Sobolev constant $C_{\text{LSI}}(\pi_{\text{data}})$, respectively Poincaré constant $C_{\text{PI}}(\pi_{\text{data}})$, the Gaussian diffusion path $(\mu_t)_{t\in[0,T]}$ satisfies for all $t \in [0, T]$

$$C_{\text{LSI}}(\mu_t) \leq \lambda_t C_{\text{LSI}}(\pi_{\text{data}}) + (1 - \lambda_t) C_{\text{LSI}}(\nu),$$
$$C_{\text{PI}}(\mu_t) \leq \lambda_t C_{\text{PI}}(\pi_{\text{data}}) + (1 - \lambda_t) C_{\text{PI}}(\nu),$$

respectively, where $C_{\text{LSI}}(\nu) = C_{\text{PI}}(\nu) = \sigma^2$.

Unlike geometric annealing (Chehab et al. (2024)), the log-Sobolev and Poincaré constants remain uniformly bounded along the entire path by the worst constant.

Introduction: Diffusion Models as Interpolations

- **Intuition**: It all boils down to finding a path of probability distributions between a simple base distribution ν and π_{data}.
- The interpolation perspective of diffusion models has been investigated by Albergo et al. (2023).
- *One-sided stochastic interpolants* exactly interpolate between ν and π_{data} by using an appropriate schedule λ_t and introducing control terms (learned as a neural network).

Introduction: Our Approach

- Practical approach to **general linear interpolation paths** between a simple distribution ν and π_{data},

$$X_t = \sqrt{\lambda_t} X + \sqrt{1 - \lambda_t} Z,$$

  where $X \sim \pi_{\text{data}}$, $Z \sim \nu$ independent of X and $\lambda_t \in [0, 1]$, $\lambda_T = 1$.

- Explore the **behaviour of Langevin dynamics driven by the gradients of** $\log \mu_t$ for $t \in [0, T]$, where μ_t are the intermediate distributions, i.e., $X_t \sim \mu_t$.

Background: Diffusion Paths

Reverse process in diffusion models = sampling along a path of probability distributions $(\mu_t)_{t\in[0,T]}$

$$\mu_t(x) = \frac{\pi_{\text{data}}(x/\sqrt{\lambda_t})}{{\lambda_t}^{d/2}} * \frac{\nu\left(x/\sqrt{1-\lambda_t}\right)}{(1-\lambda_t)^{d/2}},$$

where $*$ denotes the convolution operation, ν describes the base or *noising* distribution, and λ_t is an increasing function called schedule, such that, $\lambda_t \in [0,1]$ and $\lambda_T = 1$.

By selecting an appropriate schedule which satisfies $\lambda_0 = 0$ and $\lambda_T = 1$, the path of probability distributions $(\mu_t)_{t\in[0,T]}$ can interpolate exactly between $\mu_0 = \nu$ and $\mu_T = \pi_{\text{data}}$ in finite time.

Annealed Langevin Dynamics for Diffusion Paths

- For general diffusion paths, the "reverse process" cannot be described by a closed form SDE.

- Instead of introducing intractable control terms, we focus on **annealed Langevin dynamics** to sample from the path.

$$\mathrm{d}X_t = \nabla \log \hat{\mu}_t(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t \quad t \in [0, T/\kappa],$$

where $X_0 \sim \mu_0 = \nu$, (B_t) is a Brownian motion and $\hat{\mu}_t = \mu_{\kappa t}$, $0 < \kappa < 1$.

Annealed Langevin Dynamics for Diffusion Paths

- **Question**: How do we simulate

$$\mathrm{d}X_t = \nabla \log \hat{\mu}_t(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t \quad t \in [0, T/\kappa]?$$

- **Solution**: diffusion annealed Langevin Monte Carlo (DALMC) algorithm given by a simple Euler-Maruyama discretisation and the use of a score approximation function $s_\theta(x, t)$ (Song and Ermon (2019))

$$X_{l+1} = X_l + h_l s_\theta(X_l, t_l) + \sqrt{2h_l}\xi_l,$$

where $h_l > 0$ is the step size, $\xi_k \sim \mathcal{N}(0, I)$, $l \in \{1, \ldots, M\}$ and $0 = t_0 < \cdots < t_M = T/\kappa$ is a discretisation of the interval $[0, T/\kappa]$.

Annealed Langevin Dynamics for Diffusion Paths

- **Bad news :(**
  Even if
  $$\mathrm{d}X_t = \nabla \log \hat{\mu}_t(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t \quad t \in [0, T/\kappa]$$
  is simulated exactly, it introduces a bias, that is, $X_t \not\sim \hat{\mu}_t$

- BUT ... We quantify this bias non-asymptotically! **:)**

- A key component in determining the effectiveness of the previous dynamics will be the action of the curve $\mu = (\mu_t)_{t\in[0,T]}$ interpolating between the base distribution and π_{data}, denoted by $\mathcal{A}(\mu)$.

  Question: **What is this action exactly?**

Annealed Langevin Dynamics for Diffusion Paths

Question: **What is this action exactly?**

- The action serves as a measure of the cost of transporting ν to π_{data} along the given path (Guo et al. (2024)).
- The action of a curve of probability measures with finite second-order moment (+ some regularity conditions) is defined as follows

$$\mathcal{A}(\mu) := \int_0^T \lim_{\delta \to 0} \frac{W_2(\mu_{t+\delta}, \mu_t)}{|\delta|}.$$

- **$\mathcal{A}$ction in action**: The KL divergence between the path measure of the diffusion annealed Langevin dynamics, $\mathbb{P}_{\text{DALD}} = (p_{t,\text{DALD}})_{t\in[0,T/\kappa]}$, and that of a reference SDE such that the marginals at each time have distribution $\hat{\mu}_t$, $\mathbb{P} = (\hat{\mu}_t)_{t\in[0,T/\kappa]}$, can be bounded in terms of the action.

$\mathcal{A}(\mu)$ction in Action

Theorem

Let $\mathbb{P}_{DALD} = (p_{t,DALD})_{t\in[0,T/\kappa]}$ be the path measure of the diffusion annealed Langevin dynamics and $\mathbb{P} = (\hat{\mu}_t)_{t\in[0,T/\kappa]}$ that of a reference SDE such that $X_t \sim \hat{\mu}_t$. If $p_{0,DALD} = p_0$,

$$\mathrm{KL}(\mathbb{P}\|\mathbb{P}_{DALD}) = \frac{\kappa}{4}\mathcal{A}(\mu).$$

By the data processing inequality, we have that

$$\mathrm{KL}\left(\pi_{\text{data}}\ ||p_{T/\kappa,\text{DALD}}\right) \leq \mathrm{KL}\left(\mathbb{P}\ ||\mathbb{P}_{\text{DALD}}\right) \leq \frac{\kappa}{4}\mathcal{A}(\mu).$$

Choosing $\kappa = \mathcal{O}(\varepsilon^2/\mathcal{A}(\mu))$, we ensure $\mathrm{KL}\left(\pi_{\text{data}}\ ||p_{T/\kappa,\text{DALD}}\right) \lesssim \varepsilon^2$.

Initial Assumptions Before the Deep Dive

A1 (L^2 accurate score estimator)

The score approximation function $s_\theta(x,t)$ satisfies

$$\sum_{l=0}^{M-1} h_l \mathbb{E}_{\hat{\mu}_t}\left[\|\nabla \log \hat{\mu}_l(X_{t_l}) - s_\theta(X_{t_l}, t_l)\|^2\right] \leq \varepsilon^2_{score}.$$

where $0 = t_0 < t_1 < \cdots < t_M = T/\kappa$ is a discretisation of the interval $[0, T/\kappa]$.

A2 (Finite second-order moment of $\pi_{\mathbf{data}}$)

The data distribution π_{data} has a finite second-order moment, that is, $M_2 = \mathbb{E}_{\pi_{data}}[\|X\|^2] < \infty$.

Gaussian Diffusion Paths

Building blocks for the analysis

- **Smoothness of $(\mu_t)_t$.**

Assumption

For all $t \in [0, T]$, the scores of the intermediate distributions $\nabla \log \mu_t(x)$ are Lipschitz with finite constant L_t.

- **Bound on the action of $(\mu_t)_t$.**
  It arises naturally under some weak assumption on the schedule.

Smoothness of $(\mu_t)_t$.

Alert: The previous assumption is hard to check in general. The following assumption implies smoothness of $(\mu_t)_t$.

Assumption: Strong convexity outside of a ball

The data distribution π_{data} has density $\pi_{\text{data}} \propto e^{-V_\pi}$.

- V_π has Lipschitz continuous gradients, with Lipschitz constant L_π.
- V_π is strongly convex outside of a ball of radius r with convexity parameter $M_\pi > 0$, that is,

$$\inf_{\|x\| \geq r} \nabla^2 V_\pi \succcurlyeq M_\pi I, \quad \inf_{\|x\| < r} \nabla^2 V_\pi \succcurlyeq -L_\pi I.$$

Vacher et al. (2025) obtain alternative bounds on the Lipschitz constant L_t.

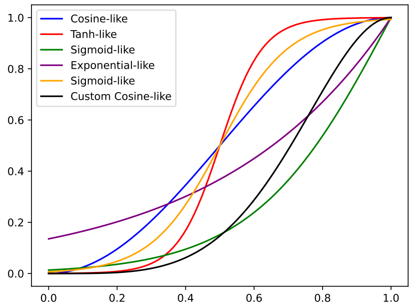
Gaussian Diffusion Paths

Action of $(\mu_t)_t$.

Assumption. (Schedule)

Let $\lambda_t : \mathbb{R}^+ \to [0, 1]$ be non-decreasing in t and weakly differentiable, such that there exists a constant C_λ satisfying either of the following conditions

$$\max_{t\in[0,T]} |\partial_t \log \lambda_t| \le C_\lambda \qquad \text{or} \qquad \max_{t\in[0,T]} \left| \frac{\partial_t \lambda_t}{\sqrt{\lambda_t(1-\lambda_t)}} \right| \le C_\lambda.$$



Gaussian Diffusion Paths

Action of $(\mu_t)_t$.

Assumption. (Schedule)

Let $\lambda_t : \mathbb{R}^+ \to [0,1]$ be non-decreasing in t and weakly differentiable, such that there exists a constant C_λ satisfying either of the following conditions

$$\max_{t\in[0,T]} |\partial_t \log \lambda_t| \leq C_\lambda \qquad \text{or} \qquad \max_{t\in[0,T]} \left| \frac{\partial_t \lambda_t}{\sqrt{\lambda_t(1-\lambda_t)}} \right| \leq C_\lambda .$$

Lemma. (Action bound)

If π_{data} has bounded second-order moment and λ_t satisfies the assumption above, the action is upper bounded by

$$\mathcal{A}_\lambda(\mu) \lesssim C_\lambda \left(\mathbb{E}_{\pi_{\text{data}}} \left[\|X\|^2 \right] + d \right) \lesssim M_2 \vee d .$$

Gaussian Diffusion Paths

Theorem

For any $\varepsilon = \mathcal{O}(\varepsilon_{\text{score}})$, and under smoothness of $(\mu_t)_t$, finite second-order moment of π_{data} and assumption on the schedule, the Gaussian DALMC algorithm initialised at $X_0 \sim \hat{\mu}_0$ requires at most

$$\mathcal{O}\left(\frac{d(M_2 \vee d)^2 L_{\max}^2}{\varepsilon^6}\right)$$

steps to approximate π_{data} to within ε^2 KL divergence, that is,

$$\mathrm{KL}(\pi_{\text{data}} \,\|\, q_{\theta,\lambda_T}) \leq \varepsilon^2,$$

assuming a sufficiently accurate score estimator.

Heavy-Tailed Diffusion Paths

We now take the base distribution to be a Student's t-distribution, $\nu \sim t(0, \sigma^2 I, \alpha)$, with tail index $\alpha > 2$

$$\nu(x) \propto \left(1 + \frac{\|x\|^2}{\alpha\sigma^2}\right)^{-(\alpha+d)/2}.$$

Bad news: The t-distribution is not a stable distribution, unlike the Gaussian family, meaning that the convolution of two t-distributions is not necessarily a t-distribution.

Heavy-Tailed Diffusion Paths

Building blocks for the analysis

- **Smoothness of $(\mu_t)_t$.**

Assumption

For all $t \in [0, T]$, the scores of the intermediate distributions $\nabla \log \mu_t(x)$ are Lipschitz with finite constant L_t.

- **Bound on the action of $(\mu_t)_t$.**
  It arises naturally under some weak assumption on the schedule.

Heavy-Tailed Diffusion Paths

Smoothness of $(\mu_t)_t$.
The following assumptions is simpler and imply smoothness of $(\mu_t)_t$.

Assumption

The data distribution π_{data} has density with respect to the Lebesgue measure.

- $\nabla \log \pi_{\text{data}}$ is Lipschitz continuous with constant L_π
- $\|\nabla \log \pi_{\text{data}}\|^2 \leq C_\pi$ almost surely.

This assumption holds when the data distribution π_{data} can be expressed as the convolution of a compactly supported measure and a t-distribution.

Heavy-Tailed Diffusion Paths

Action of $(\mu_t)_t$.

Assumption. (Schedule)

Let $\lambda_t : \mathbb{R}^+ \to [0,1]$ be non-decreasing in t and weakly differentiable, such that there exists a constant C_λ satisfying

$$\max_{t\in[0,T]} \left| \frac{\partial_t \lambda_t}{\sqrt{\lambda_t(1-\lambda_t)}} \right| \leq C_\lambda .$$

Lemma. (Action bound)

If π_{data} has bounded second-order moment and λ_t satisfies the assumption above, the action is upper bounded by

$$\mathcal{A}_\lambda(\mu) \leq \frac{C_\lambda \pi}{8} \left(\mathbb{E}_{\pi_{\text{data}}} \left[\|X\|^2 \right] + \frac{\sigma^2 d\alpha}{\alpha - 2} \right) .$$

Heavy-Tailed Diffusion Paths

Theorem

Let $\nu \sim t(0, \sigma^2 I, \alpha)$ with $\alpha > 2$. For any $\varepsilon = \mathcal{O}(\varepsilon_{\text{score}})$, and under smoothness of $(\mu_t)_t$, finite second-order moment of π_{data} and assumption on the schedule, the heavy-tailed DALMC algorithm initialised at $X_0 \sim \hat{\mu}_0$ requires at most

$$\mathcal{O}\left(\frac{d(M_2 \vee d)^2 L_{\max}^2}{\varepsilon^6}\right)$$

steps to approximate π_{data} to within ε^2 KL divergence, that is,

$$\mathrm{KL}(\pi_{\text{data}} \,\|\, q_{\theta,\lambda_T}) \leq \varepsilon^2,$$

assuming a sufficiently accurate score estimator.

Remark: same upper bound for the complexity as in the Gaussian case

Some Final Remarks

Take home messages:

- We have obtained non-asymptotic guarantees in KL divergence for the DALMC algorithm when the base distribution is either Gaussian or Student's t.
- In our paper, we also obtain bounds when replacing the assumption on the smoothness of $(\mu_t)_t$ with a weaker assumption $\mathbb{E}_{\pi_{\text{data}}} \|\nabla V_\pi(X)\|^8 \leq K_\pi^2$.

Some future directions:

- Developing more efficient numerical schemes, reducing dimensional dependencies in error bounds, and applying this framework to other generative models.

Thank you!