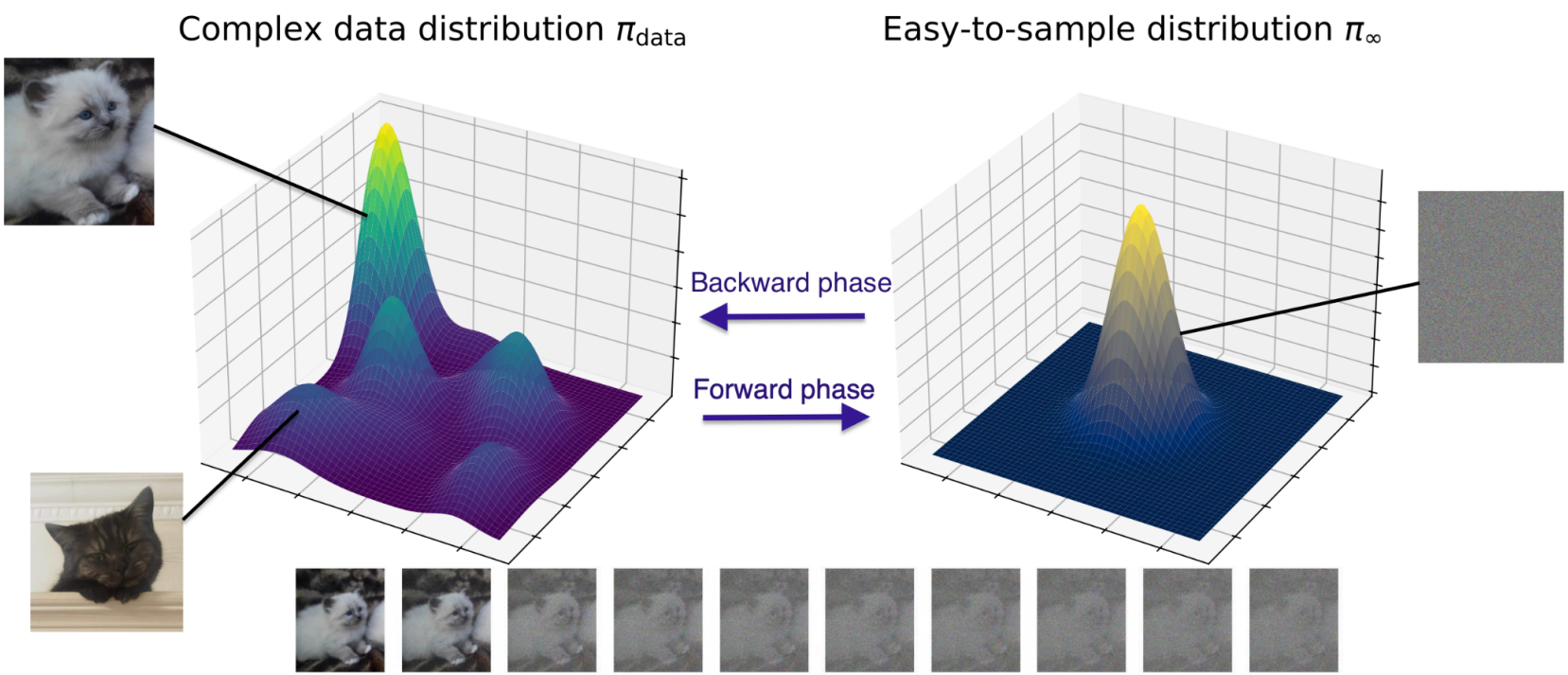

# Wasserstein Convergence of Critically Damped Langevin Diffusions

**Stanislas Strasman**, Sobihan Surendran, Claire Boyer, Sylvain Le Corff, Vincent Lemaire, **Antonio Ocello**.

NeurIPS 2025


# Generative Modeling Framework

- $\mathcal{D} = \{x_i\}_{i=1}^n \in (\mathbb{R}^d)^n$ a collection of i.i.d. samples from an unknown distribution $\pi_{\text{data}}$.
- Goal: **generate new samples from** $\pi_{\text{data}}$ (i.e. find a proba $\pi_\infty$ and a simulable kernel $Q$ such that $\pi_{\text{data}} \approx \pi_\infty Q$).



# Forward-backward SDEs

- $\left(\overrightarrow{X}_t\right)_{t\in[0,T]}$ is solution to an **Ornstein–Uhlenbeck** process:

$$\mathrm{d}\overrightarrow{X}_t = -\overrightarrow{X}_t \mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t, \quad \overrightarrow{X}_0 \sim \pi_{\mathrm{data}}\,.$$

- This forward process admits a **time-reversed process**, i.e. $\left(\overleftarrow{X}_t\right)_{t\in[0,T]} \overset{\mathcal{L}}{=} \left(\overrightarrow{X}_{T-t}\right)_{t\in[0,T]}$ with,

$$\mathrm{d}\overleftarrow{X}_t = \left(\overleftarrow{X}_t + 2\underbrace{\nabla \log p_{T-t}\left(\overleftarrow{X}_t\right)}_{\textbf{score function}}\right)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t\,, \quad \overleftarrow{X}_0 \sim p_T$$

  with $p_t$ the p.d.f. of $\overrightarrow{X}_t$ and semigroup $Q_t$ associated to $\overleftarrow{X}_t$.

- This is (**almost**) a **generative model** since $\pi_{\mathrm{data}} \overset{\mathcal{L}}{=} p_T Q_T$.

# Learning the score function

- Train a **deep neural network** $s_\theta : [0, T] \times \mathbb{R}^d \mapsto \mathbb{R}^d$ to minimize:

$$\mathcal{L}_{\text{naive}}(\theta) = \mathbb{E}\left[\left\| s_\theta\left(\tau, \overrightarrow{X}_\tau\right) - \nabla \log p_\tau\left(\overrightarrow{X}_\tau\right)\right\|^2\right],$$

  with $\tau \sim \mathcal{U}(0, T)$ independent of the forward process $(\overrightarrow{X}_t)_{t\geq 0}$.

- But $p_\tau(x)$ is a convolution between a Gaussian kernel and the **unknown** $\pi_{\text{data}}$ !
- 💡 its **conditional version** shares the same optimum

$$\mathcal{L}_{\text{score}}(\theta) = \mathbb{E}\left[\| s_\theta\left(\tau, \overrightarrow{X}_\tau\right) - \nabla \log p_\tau\left(\overrightarrow{X}_\tau | \overrightarrow{X}_0\right) \|^2\right]$$

with,

$$\nabla \log p_\tau(\overrightarrow{X}_\tau | \overrightarrow{X}_0) = \frac{e^{-\tau}\overrightarrow{X}_0 - \overrightarrow{X}_\tau}{1 - e^{-2\tau}}.$$

# CLD: Extending the Phase Space of SGMs

**Forward process:**

- Augment data space with a **velocity component** $\overrightarrow{\mathbf{U}}_t = (\overrightarrow{X}_t, \overrightarrow{V}_t)^\top \in \mathbb{R}^{2d}$ and couple $X_t$ and $V_t$ through *Hamiltonian-like* interactions.
- **Noise injection** $B_t \in \mathbb{R}^{2d}$ only on the **velocity** component.

$$\mathrm{d}\overrightarrow{\mathbf{U}}_t = A\overrightarrow{\mathbf{U}}_t \mathrm{d}t + \Sigma \mathrm{d}B_t\,, \quad \overrightarrow{\mathbf{U}}_0 \sim \pi_{\text{data}} \otimes \pi_v \tag{1}$$

with

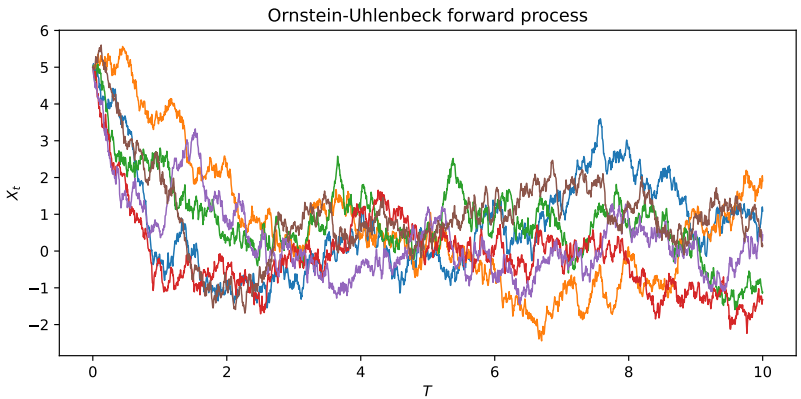
$$A = \begin{pmatrix} 0 & a^2 \\ -1 & -2a \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 0 & 0 \\ 0 & \sigma \end{pmatrix} \text{ and } \pi_v \sim \mathcal{N}(\mathbf{0}_d, v^2)\,.$$

**Backward process:** This leads to the backward process:

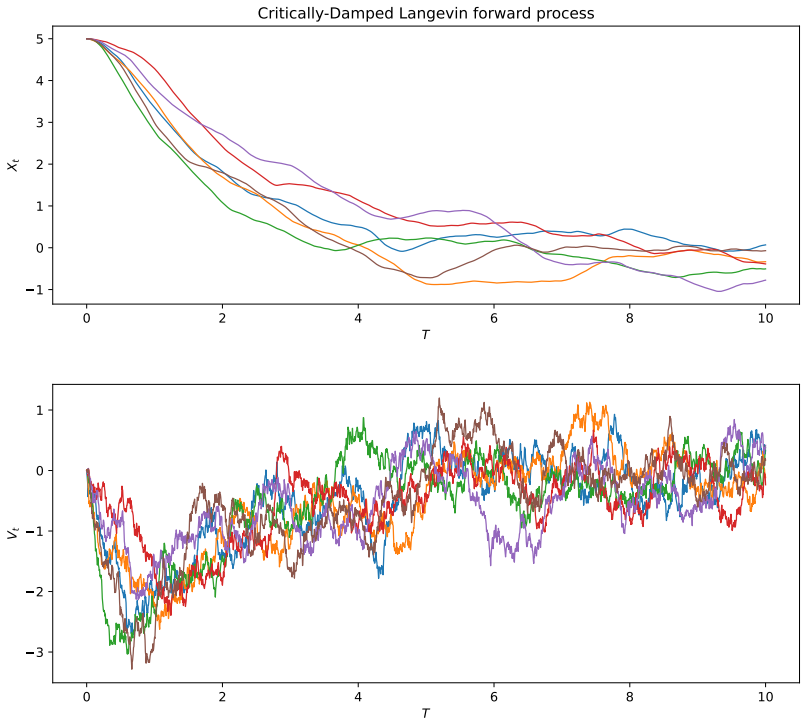
$$\mathrm{d}\overleftarrow{\mathbf{U}}_t = -A\overleftarrow{\mathbf{U}}_t \mathrm{d}t + \Sigma^2 \nabla \log p_{T-t}\left(\overleftarrow{\mathbf{U}}_t\right)\mathrm{d}t + \Sigma \mathrm{d}B_t\,,$$

where $p_t$ denotes p.d.f. of (1).

# Noising Process Comparison




**OU process**




**CLD process**

**(position & velocity)**

# Learning the score function (the CLD case)

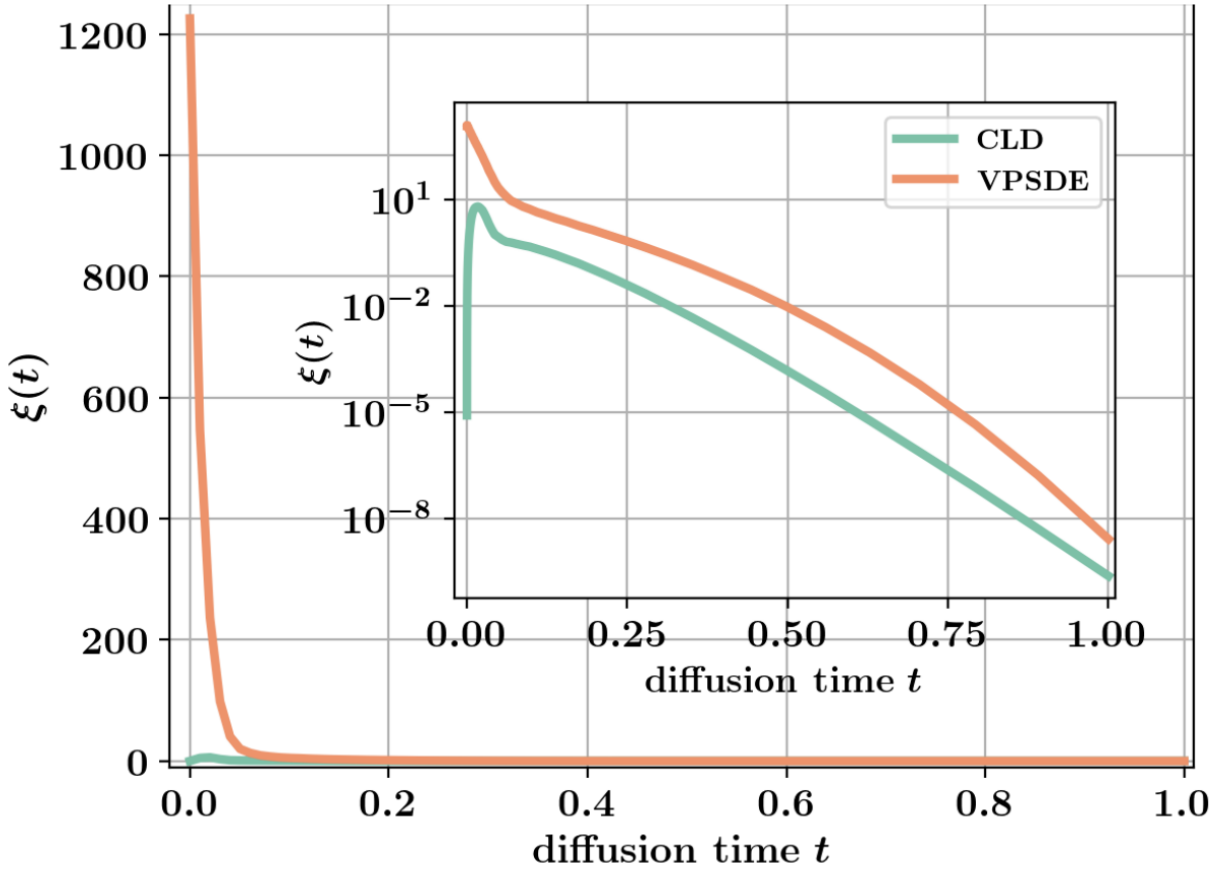
- As before $s_\theta$ trained to learn conditional score but on the whole phase-space state $\overrightarrow{\mathbf{U}}_t = (\overrightarrow{X}_t, \overrightarrow{V}_t)$:

$$\mathcal{L}_{\mathrm{DSM}}(\theta) = \mathbb{E}\left[\left\| s_\theta(t, \overrightarrow{\mathbf{U}}_t) - \nabla \log p_t(\overrightarrow{\mathbf{U}}_t \mid \overrightarrow{\mathbf{U}}_0)\right\|^2\right] .$$

- However, we know that $\overrightarrow{V}_0 \sim \mathcal{N}(0, v^2)$, so we can marginalize $\overrightarrow{\mathbf{U}}_0 = (\overrightarrow{X}_0, \overrightarrow{V}_0)^\top$ over $V_0$, leading to a **closed-form expression** of $\nabla \log p_t(\overrightarrow{\mathbf{U}}_t \mid \overrightarrow{X}_0)$:

$$\mathcal{L}_{\mathrm{HSM}}(\theta) = \mathbb{E}\left[\left\| s_\theta(t, \overrightarrow{\mathbf{U}}_t) - \nabla \log p_t(\overrightarrow{\mathbf{U}}_t \mid \overrightarrow{X}_0)\right\|^2\right] ,$$

# HSM yields **more stable training** objective.




$\xi(t) = L^2$–difference between diffused-data score and Normal score.

Figure from Dockhorn, Vahdat & Kreis (2022), *Score-Based Generative Modeling with Critically-Damped Langevin Diffusion*.

# Sources of Error in CLD-SGMs I

- Let $Q_t(x, \mathrm{d}y) = \mathbb{P}\left(\overleftarrow{\mathbf{U}}_t \in \mathrm{d}y | \overleftarrow{\mathbf{U}}_0 = x\right)$, **time-reversal** implies

$$\pi_{\text{data}} \overset{\mathcal{L}}{=} p_T Q_T \,.$$

- $p_T$ is not directly accessible. But for large $T$, the process forgets its initialization, $\overrightarrow{\mathbf{U}}_t = (\overrightarrow{X}_t, \overrightarrow{V}_t)^\top \in \mathbb{R}^{2d}$ as

$$\overrightarrow{\mathbf{U}}_t = \mathrm{e}^{tA}\overrightarrow{\mathbf{U}}_0 + \int_0^t \mathrm{e}^{(t-s)A}\Sigma \mathrm{d}B_s \tag{2}$$

and converges to $\pi_\infty \sim \mathcal{N}(\mathbf{0}_{2d}, \Sigma_\infty)$.

$$p_T \approx \pi_\infty$$

⚠ **Mixing-time error:** $\pi_{\text{data}} \approx \pi_\infty Q_T$

# Sources of Error in CLD-SGMs II

- Score function $\nabla \log p_t$ is intractable but can be approximated by a deep neural network $s_\theta$ via score matching.

$$\|s_\theta(t, \mathbf{U}_t) - \nabla \log p_t(\mathbf{U}_t)\|_{L_2} \leq M$$

⚠ **Approximation error:** $\pi_{\text{data}} \approx \pi_\infty Q_T^\theta$

- Backward drift is **non-linear** and should be discretized into $N$ finite steps (Euler–Maruyama or symplectic integrators).

⚠ **Discretization error**: $\pi_{\text{data}} \approx \pi_\infty Q_{T,N}^\theta := \widehat{\pi}_{\infty,N}^\theta$

# $\mathcal{W}_2$ Upper Bound

$$\mathcal{W}_2(\pi_{\text{data}}, \widehat{\pi}^{\theta}_{\infty,N}) \leq \underbrace{\mathcal{W}_2(\mathcal{L}(\overleftarrow{\mathbf{U}}_T), \mathcal{L}(\bar{\mathbf{U}}_N)}_{\text{Discretization}} + \underbrace{\mathcal{W}_2(\mathcal{L}(\bar{\mathbf{U}}_N), \mathcal{L}(\bar{\mathbf{U}}_{\infty,N})}_{\text{Mixing time}}$$
$$+ \underbrace{\mathcal{W}_2(\mathcal{L}(\bar{\mathbf{U}}_{\infty,N}), \mathcal{L}(\bar{\mathbf{U}}^{\theta}_{\infty,N}))}_{\text{Score approximation}},$$

where $T > 0$ denotes the diffusion time horizon and $N$ the number of discretization steps.

$\Rightarrow$ Standard proofs that control these errors using the **strong log-concavity of** $p_t$ fail in this setting, since noise is injected only into the velocity component, making the SDEs **hypoelliptic**.
$\Rightarrow$ Two solutions were developed to study this kinetic algorithm.

# Solution 1: Long-term regularity of the renormalized score

**Idea:** Introduce a *renormalized* formulation of the backward process:

$$\mathrm{d}\overleftarrow{\mathbf{U}}_t = \tilde{A}\overleftarrow{\mathbf{U}}_t \mathrm{d}t + \Sigma^2 \nabla \log \tilde{p}_{T-t}(\overleftarrow{\mathbf{U}}_t)\mathrm{d}t + \Sigma \mathrm{d}B_t, \qquad \tilde{p}_t := \frac{p_t}{p_\infty}.$$

**Key properties:**

1. $\tilde{p}_t$ "quantifies" **deviation from equilibrium** $p_\infty$.
2. Its curvature $\nabla^2 \log \tilde{p}_t$ characterizes the **regularity of the score**, for all $t \in (0, T]$,

$$\left\| \nabla^2 \log \tilde{p}_t(\cdot) \right\| \leq C \left(1 + \frac{1}{\sqrt{t}}\right) \mathrm{e}^{-2at} = \tilde{L}_t \,.$$

3. Recover a bound of the type, as for general SGMs

$$\mathcal{W}_2(\pi_{\mathrm{data}}, \widehat{\pi}^{\theta}_{\infty,N}) \leq \mathrm{e}^{-T} c_1 + M c_2 + \sqrt{T/N} c_3 \,.$$

# Solution 2: Restoring ellipticity

**Idea:** Inject a small amount of noise into *all* coordinates:

$$\Sigma = \begin{pmatrix} \varepsilon & 0 \\ 0 & \sigma \end{pmatrix}, \qquad \varepsilon > 0.$$

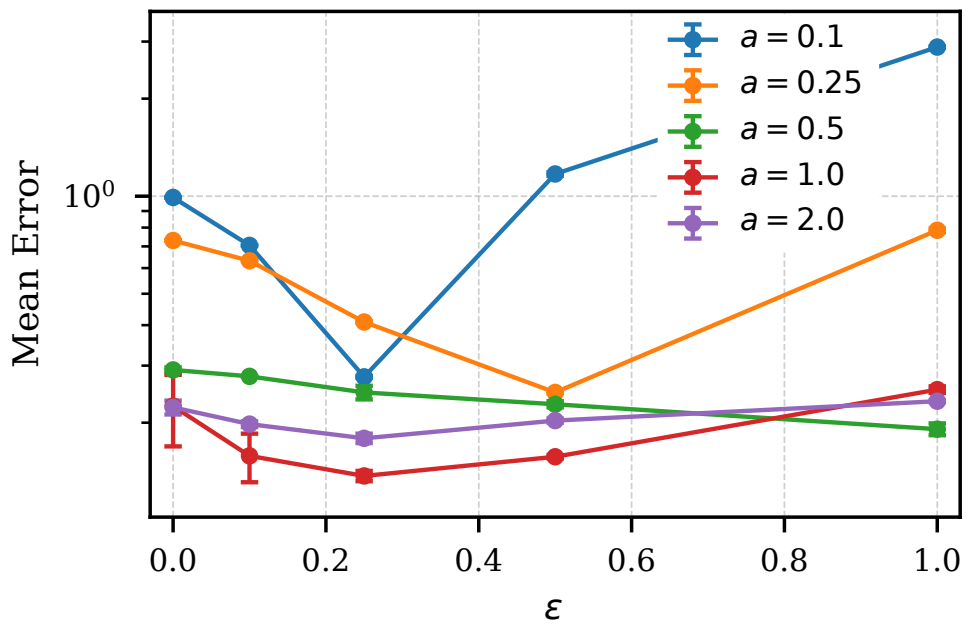
**Consequences:**

- **Uniform ellipticity:** (multi-dimensional O.U. structure).
- **More quantitative bounds:** standard log-concave tools apply.
- **In practice:** provides a new parameter $\varepsilon$ to control the regularity of the sample paths.

# Numerical Illustration

**Empirical results (Funnel dataset, $d = 100$).**

- Small $\varepsilon$ often **improves** sliced-$\mathcal{W}_2$ vs. $\varepsilon = 0$ (CLD baseline).
- **Trade-off:** training becomes more expensive, since the network must learn full gradients.




Mean $\mathcal{W}_2$ over 5 runs; error bars represent $\pm$ one standard deviation.