Score-Based Generative Modeling with Critically-Damped Langevin Diffusion

Dockhorn, Vahdat, and Kreis (2022)

Zachary Berens

April 8, 2024

Background & Related Work

Remark

"Creating noise from data is easy; creating data from noise is generative modeling." (Song et al. 2021a)

Recall that the goal of score-based generative modeling is to learn the score of the distribution that generates our data: $\nabla_x \log p_*(x)$. Doing so permits approximation of $p_*(x)$ in the following sense:

Theorem

The Fisher divergence $\frac{1}{2}\mathbb{E}_*||\nabla_x \log p(x) - \nabla_x \log p_*(x)||_2^2$ *is equal to* 0 *if and only if* $p = p_*$ *where the expectation is taken w.r.t. the true distribution.*

Fisher divergence is an alternative to KL-divergence for measuring the closeness of two distributions. In our case, it appears as the natural thing to minimize (as I'll explain).

Background & Related Work

We saw that

$$\frac{1}{2}\mathbb{E}_*||s_\theta(x) - \nabla_x \log p_*(x)||_2^2 = \mathbb{E}_* \left[\text{tr}(\nabla_x s_\theta(x)) + \frac{1}{2}||s_\theta(x)||_2^2\right]$$

up to a constant. But score matching is not scalable to deep and high-dimensional settings because computing $\text{tr}(\nabla_x s_\theta(x))$ becomes intractable. One solution we saw was "sliced score matching" where we use random projections to approximate $\text{tr}(\nabla_x s_\theta(x))$. There is another solution: *denoising score-matching*. Instead of worrying about $\text{tr}(\nabla_x s_\theta(x))$, we instead perturb x according to some noise distribution $p_\sigma(\tilde{x}|x)$. We then use score matching estimate the score of the *perturbed distribution* $p_\sigma(\tilde{x}) := \int p_\sigma(\tilde{x}|x)p_*(x)dx$.

Theorem (Vincent, 2011)

The objective is equivalent to

$$\frac{1}{2}\mathbb{E}_{p_\sigma(\tilde{x}|x)p_*(x)}\left[||s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x)||_2^2\right]$$

Background & Related Work

Denoising Score Matching w/ Langevin Dynamics: We define a discrete-time forward diffusion process. Select

$$\sigma_{\text{min}} = \sigma_1 < \ldots < \sigma_N = \sigma_{\text{max}}$$

such that $p_{\sigma_1} \approx p_*(x)$ and $p_{\sigma_N} \approx \mathcal{N}(0, \sigma_N^2 I)$. Define

$$p_\sigma(\tilde{x}|x) := \mathcal{N}(\tilde{x}; x, \sigma^2 I)$$

"Noise Conditional Score Network" (Song and Ermon, 2019). Take a weighted sum of denoising score matching objectives following the previous theorem.

$$\theta^* = \underset{\theta}{\text{argmin}} \sum_{i=1}^{N} \sigma_i^2 \mathbb{E}_{p_*(x)} \mathbb{E}_{p_{\sigma_i}(\tilde{x}|x)} [||s_\theta(\tilde{x}, \sigma_i) - \nabla_{\tilde{x}} \log p_{\sigma_i}(\tilde{x}|x)||_2^2]$$

Use MALA to get a sample for each p_{σ_i} sequentially. Start with some $x_N \sim \mathcal{N}(x|0, \sigma_N^2 I)$. With enough samples and small enough step size, we can basically sample from $p_{\sigma_1} \approx p_*$.

Background & Related Work

Perturbing data with multiple noise scales is key. What if we used infinitely many noise scales such that the noisy data evolves according to an SDE? We now have a continuous time diffusion process $\{x(t)\}_{t=0}^{T}, t \in [0, T]$ such that $x(T) \sim p_T$ is some tractable prior (e.g., Gaussian). The diffusion will be given by (the solution to) an Itô SDE: $dx = f(x, t)dt + g(t)d\mathbf{w}$.

$$f(\cdot, t) : \mathbb{R}^d \longrightarrow \mathbb{R}^d \text{ "drift coefficient"}$$

$$g(\cdot, t) : \mathbb{R}^d \longrightarrow \mathbb{R}^d \text{ "diffusion coefficient"}$$

$$\mathbf{w} : \text{ Wiener process (Brownian motion)}$$

Then we have a transition kernel from $x(s)$ to $x(t)$ with $0 \leq s < t \leq T$ given by $p_{st}(x(t)|x(s))$. Under suitable conditions (smooth, unique solution the Kolmogorov equation; Anderson, 1982), a reverse SDE exists. We'd like to learn this reverse-time SDE so we can "undiffuse" samples from the prior $x(T)$ to get samples from $p_*(x)$.

Background & Related Work

The cool thing about diffusion models is that learning the reverse SDE is equivalent to learning the score at each noise scale! In particular, learning the reverse SDE is equivalent to learning the score:

Theorem (Anderson, 1982)

The reverse-time of SDE of the above Itô SDE is given by

$$dx = [f(x,t) - g(x,t)g(x,t)^T \nabla_x \log p_t(x)]dt + g(t)\overline{\mathbf{w}}$$

where t goes from T to 0 and $\overline{\mathbf{w}}$ denotes the standard Wiener process when time flows backwards. And dt is an infinitesimal negative time step.

The idea is that we want $\nabla_x \log p_t(x)$ for all t. We then get the following objective (Song et al, 2021b): take argmin_θ of

$$\mathbb{E}_{t\sim\mathcal{U}[0,T]}\left\{\lambda(t)\mathbb{E}_{x(0)}\mathbb{E}_{x(t)|x(0)}\left[||s_\theta(x(t),t) - \nabla_{x(t)} \log p_{0t}(x(t)|x(0))||_2^2\right]\right\}$$

Where $\lambda(t) : [0, T] \longrightarrow \mathbb{R}_{>0}$ is a regularization term. I'll note that we can recover discrete-time denoising diffusion as discretizations of this setup.

Critically-Damped Langevin Diffusion

Innovations:

1. State-space augmentation for smooth denoising.
2. Hybrid score matching.
3. Simplification of denoising by using a more complex diffusion process.
4. Easier objective than learning scores directly.
5. Novel, high-quality sampler for SDE's of this sort (outperforms Euler-Maruyama for example).

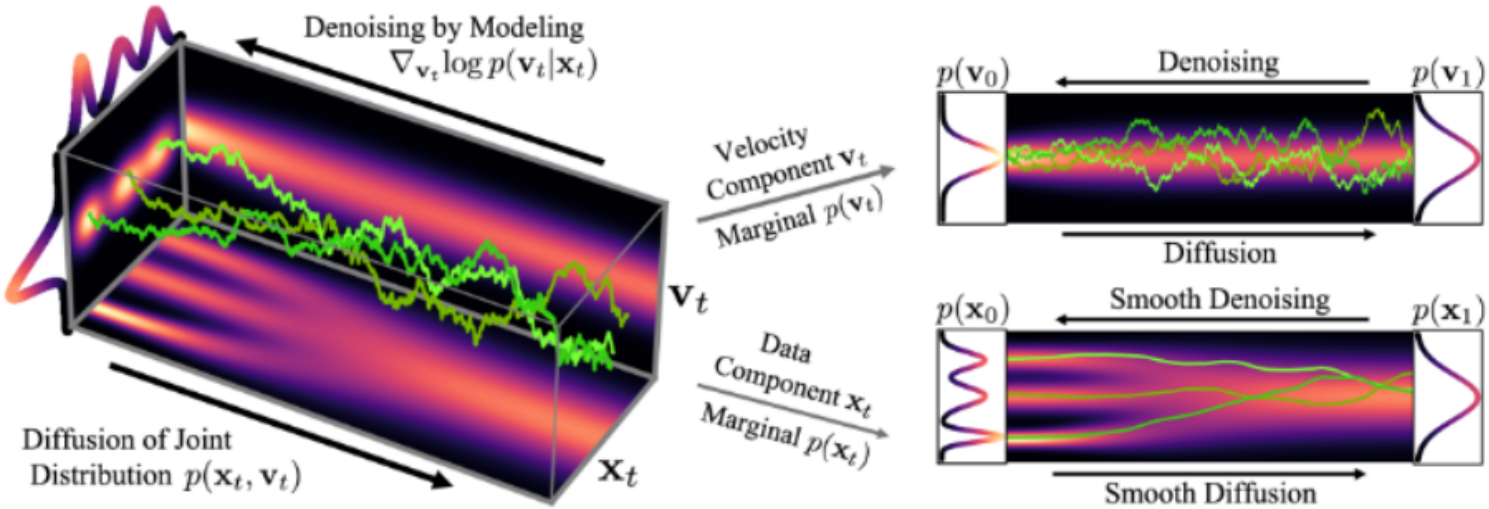


Figure 1: Dockhorn, Vahdat, and Kreis (2022)

State-Space Augmentation

We begin by augmentung data $x_t \in \mathbb{R}^d$ with a "velocity" $v_t \in \mathbb{R}^d$. We only put noise into v_t. Define $u_t = \left[x_t, v_t\right]^T \in \mathbb{R}^{2d}$. We then let

$$f(u_t, t) = \left(\begin{pmatrix} 0 & \beta M^{-1} \\ -\beta & -\Gamma \beta M^{-1} \end{pmatrix} \right) \otimes I_d$$

$$G(u_t, t) = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{2\Gamma\beta} \end{pmatrix} \otimes I_d$$

Where β is some scalar that we use to rescale the diffusion process for convergence in $[0, T]$, Γ is a "friction" coefficient (this will become clear in a moment), and M is the "mass" which is a hyperparameter measuring the coupling between x and v. This yields the SDE:

$$\begin{pmatrix} dx_t \\ dv_t \end{pmatrix} = \begin{pmatrix} M^{-1} v_t \\ -x_t \end{pmatrix} \beta dt + \begin{pmatrix} 0_d \\ -\Gamma M^{-1} v_t \end{pmatrix} \beta dt + \begin{pmatrix} 0 \\ \sqrt{2\Gamma\beta} \end{pmatrix} d\mathbf{w}_t$$

The first term is a "Hamiltonian" term whereas the sum of the second and third terms is an "Ornstein-Uhlenbeck" term. Each x_i is independently coupled to a velocity v_i.

Damping

- $\Gamma^2 < 4M$ is *Underdamped* $\implies$ Hamiltonian dominates $\implies$ x_t and v_t oscillate, slowing down the convergence.
- $\Gamma^2 > 4M$ is *Overdamped* $\implies$ the Ornstein-Uhlenbeck term dominates $\implies$ there's slow convergence because of too much noise.
- $\Gamma^2 = 4M$ is *Critically Damped* $\implies$ just right! (insert Goldilocks reference)

Where do these terms come from? Physics. Consider the oscillation of a spring mass.

- Underdamped means that the oscillation keeps overshooting the equilibrium.
- Overdamped happens if e.g., the spring is in a viscous fluid
- Critically Damped means that you return to equilibrium in the minimal amount of time.

Damping

Damped harmonic oscillator (ζ over, under, or equal to 1):

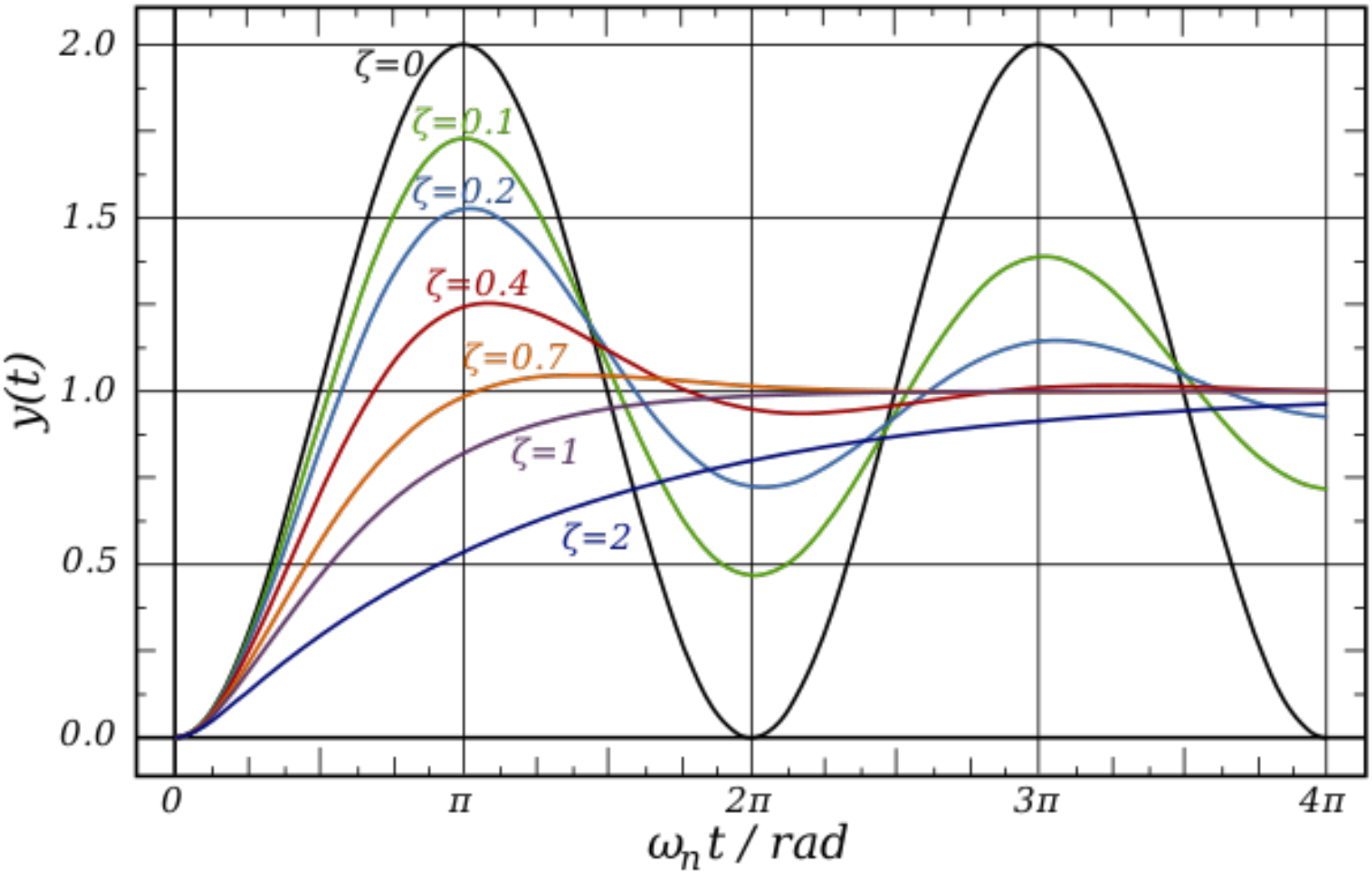


Figure some guy uploaded to wikipedia.

Damping

Idea: Critically-Damped Langevin Diffusion (CLD) is an instance of general Langevin diffusion which is given by

$$\begin{pmatrix} dx \\ dv \end{pmatrix} = \begin{pmatrix} M^{-1}v \\ \nabla_x \log p_{pot}(x) \end{pmatrix} \beta dt + \begin{pmatrix} 0_d \\ -\Gamma M^{-1} v \end{pmatrix} \beta dt + \begin{pmatrix} 0 \\ \sqrt{2\Gamma\beta} \end{pmatrix} d\mathbf{w}$$

Such Langevin diffusion is known to have equilibrium given by $p_{pot}(x)\mathcal{N}(v; 0_d, I_d)$. In our case, $p_{pot} = \mathcal{N}(x; 0_d, I_d)$. Hence, the equilibrium solution corresponds to the classical harmonic oscillator. We'd like to diffuse to this equilibrium.

Objective

For our forward process, begin with $p_0(u_0) = p(x_0)p(v_0) = p_*(x_0)\mathcal{N}(v_0; 0_d, \gamma M I_d)$ where $\gamma < 1$ is a hyperparameter ("initial velocity distribution width"). We diffuse to the equilibrium $p_{EQ}(u) = \mathcal{N}(x; 0_d, I_d)\mathcal{N}(v; o_d, M I_d)$. The objective ends up being very similar to the one on slide 6. It's given by

$$\mathbb{E}_{t\sim\mathcal{U}[0,T]}\mathbb{E}_{u_t\sim p_t(u_t)}\left[\lambda(t)||s_\theta(u_t, t) - \nabla_{v_t}\log p_t(u_t)||_2^2\right]$$

The important thing here is that we only need to learn the score with respect to v_t, which is a direct consequence of the fact that we only inject noise into v_t. Moreover, we need only learn the score of $p_t(v_t|x_t)$ which is arguably easier than learning the score of $p_t(x_t)$ or, a fortiori, $p_t(u_t)$. Indeed, WLOG:

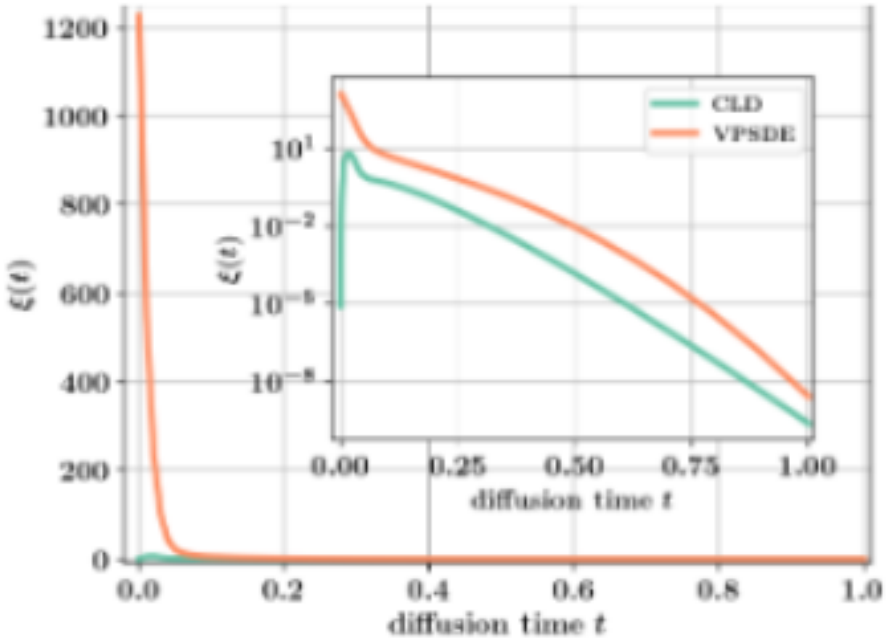
$$p_t(u_t) = p_t(x_t, v_t) = p_t(v_t|x_t)p_t(x_t)$$

Hence

$$\begin{aligned}\nabla_{v_t}\log p_t(u_t) &= \nabla_{v_t}[\log p_t(v_t|x_t) + \log p_t(x_t)] \\ &= \nabla_{v_t}\log p_t(v_t|x_t)\end{aligned}$$

Objective

To see why learning this would be easier, we note first that the velocity is initialized from a simple normal distribution such that $p_t(v_t|x_t)$ is closer to normal $\forall t \geq 0$ than $p_t(x_t)$ is. Starting e.g., at $t = 0$ we have $p(x) \perp\!\!\!\perp p(v)$ so $p_0(v_0|x_0) = p_0(v_0)$ is normal. That this continues can be shown empirically.



Difference $\xi(t)$ (via L2 norm) between score of diffused data and score of Normal distribution. Figure 2, *ibid.*

Hybrid Score Matching

Notice that we can't train the above objective directly because we don't have access to $p_t(u_t)$. We could do denoising score matching. But $t = 0$ has a complex data distribution. So instead, we resort to *Hybrid Score Matching* (HSM). The idea is to sample from $p_0(x_0) \approx p_*(x_0)$ but diffuse *while marginalizing over the full initial velocity distribution* $p_0(v_0) = \mathcal{N}(v; 0_d, \gamma M I_d)$. Since $p_0(v_0)$ is normal and f and G are affine, $p(u_t|x_0)$ is normal and so it's tractable. The HSM objective is thus given by

$$\mathbb{E}_{t\in[0,T]}\mathbb{E}_{x\sim p_0(x_0)}\mathbb{E}_{u_t\sim p_T(u_t|x_0)}[\lambda(t)||s_\theta(u_t,t) - \nabla_{v_t}\log p_t(u_t|x_0)||_2^2]$$

One can show that HSM has less variance than DSM in practice.

Hybrid Score Matching

Heuristically, $\mathbb{E}_{p_0(v_0)}$ is analytically solved whereas DSM would have to use samples. Empirically, we see a marked difference in variances

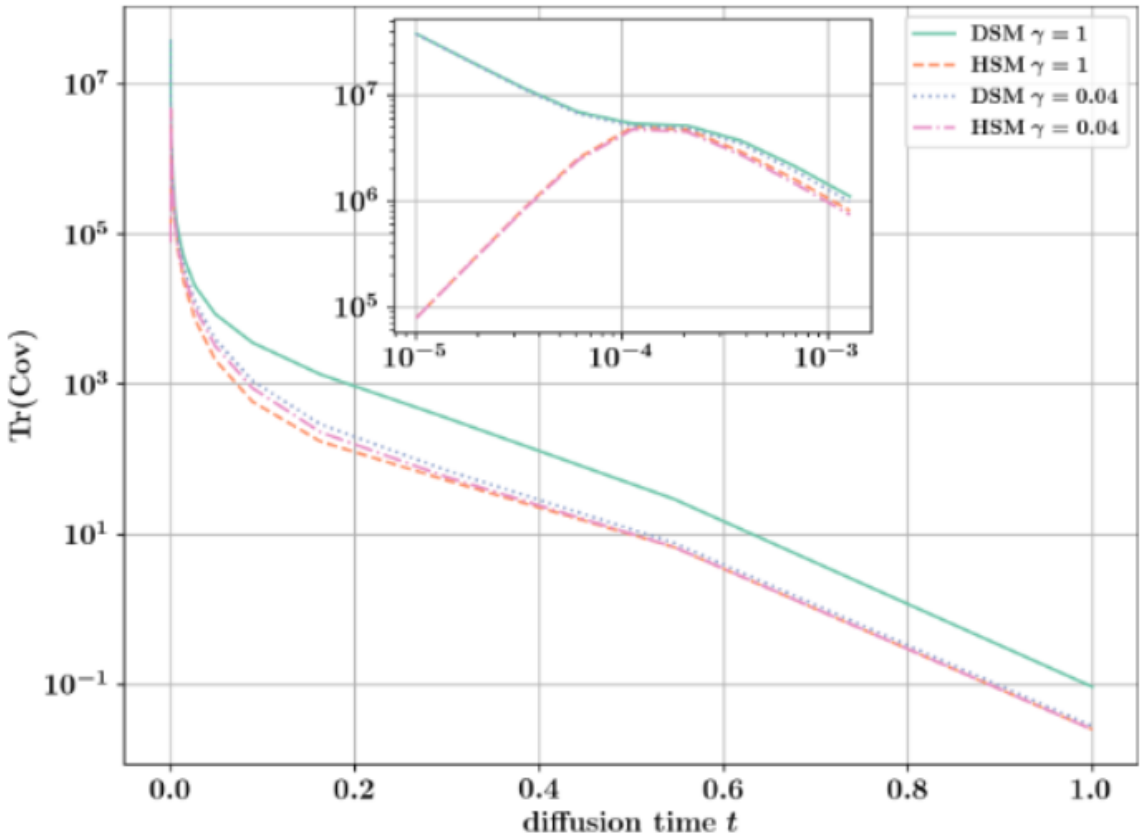


Figure 7: Traces of the estimated covariance matrices.

From *ibid.*

Symmetric Splitting CLD Sampler

The authors construct a new SDE solver (integrator) tailored to CLD. The construction is quite technical, so we'll only give a high-level overview. We have the generative reverse process

$$\begin{pmatrix} d\overline{x}_t \\ d\overline{v}_t \end{pmatrix} = \begin{pmatrix} -M^{-1}\overline{v}_t \\ \overline{x}_t \end{pmatrix} \beta dt + \begin{pmatrix} 0_d \\ -\Gamma M^{-1}\overline{v}_t \end{pmatrix} \beta dt + \begin{pmatrix} 0_d \\ \sqrt{2\Gamma\beta} d\mathbf{w}_t \end{pmatrix}$$

$$+ \begin{pmatrix} 0_d \\ 2\Gamma[s(\overline{u}_t, T-t) + M^{-1}\overline{v}_t] \end{pmatrix} \beta dt$$

The first term corresponds to the Hamiltonian term, the second and third are the Ornstein-Uhlenbeck term, and the last term corresponds to the score. The Fokker-Planck equation is a partial differential equation that describes the time evolution of the probability density function of a stochastic process. We use this formalism. We can formally construct a solution $\overline{u}_t = e^{t(A+B)}\overline{u}_0$ where A and B are the Fokker-Planck operators with A corresponding to the Hamiltonian + Ornstein-Uhlenbeck contribution and B corresponding to the score. Physics enjoyers will recognize this as the propagator.

Symmetric Splitting CLD Sampler

We then apply Baker-Campbell-Hausdorff (Fokker-Planck operators don't commute) to see that, for large N, discretizing the dynamics to steps of size t/N allows us to apply the individual e^A's and e^B's consecutively. Then we can approximate the score term using Euler-Maruyama.