SCHEDULE-FREE DENOISING DIFFUSION SAMPLING USING TIME SCORES

Anonymous authors
Paper under double-blind review

ABSTRACT

Annealing-based neural samplers amortize sampling from unnormalized target distributions by transporting samples along a family of intermediate smoothed densities connecting a tractable source distribution to the target. Existing approaches typically rely on a predefined noise schedule, prescribing the progression through the intermediate densities independently of the current sample state. To remedy this, we propose Dual Score Diffusion Sampling (DSDS), which instead treats the diffusion time, particularly its associated noise varaince, as a stochastic variable and lifts the sampling problem to the joint space. In this lifted space, both variables evolve jointly through stochastic dynamics, which allows the sampler to adaptively navigate the joint probability space. Across synthetic and image-generation experiments, we show that this leads to improved robustness, while remaining competitive with schedule-based samplers.

1 INTRODUCTION

We consider the problem of sampling from a probability distribution on $\mathbb{R}^d$ with Gibbs-Boltzmann density

$$p_{\mathrm{data}}(\boldsymbol{x}) = \frac{\exp(-U(\boldsymbol{x}))}{Z}, \qquad Z = \int_{\mathbb{R}^d} \exp(-U(\boldsymbol{\xi}))\,\mathrm{d}\boldsymbol{\xi}, \tag{1}$$

where $U : \mathbb{R}^d \to \mathbb{R}$ is the potential function and Z is the partition function. For example, p_{data} may be available analytically, as for a Gaussian mixture model, or approximated from data by a learned model $p_\theta \approx p_{\mathrm{data}}$. Sampling from such distributions is a fundamental task in machine learning (Welling & Teh, 2011; Teh et al., 2016), generative modeling (Song & Ermon, 2019; Ho et al., 2020), and Bayesian inverse problems (Stuart, 2010; Martin et al., 2012). In high-dimensional settings, exact sampling is typically unavailable, and one instead resorts to Markov chain Monte Carlo (MCMC) methods, which can operate without knowledge of the normalizing constant Z.

A prominent example is overdamped Langevin dynamics,

$$\mathrm{d}\boldsymbol{x}_s = \nabla_{\boldsymbol{x}} \log p_{\mathrm{data}}(\boldsymbol{x}_s)\,\mathrm{d}s + \sqrt{2}\,\mathrm{d}\boldsymbol{w}_s, \tag{2}$$

with standard Wiener process $\boldsymbol{w}_s$. Under suitable regularity and ergodicity conditions, p_{data} is the invariant density of this diffusion, such that the law of $\boldsymbol{x}_s$ converges to p_{data} as $s \to \infty$ (Roberts & Tweedie, 1996). Despite its simplicity, Langevin dynamics can mix slowly for high-dimensional or multimodal target distributions, as trajectories may remain trapped in regions of high probability and transition only rarely between separated modes.

Annealing provides an effective mechanism for alleviating this problem. In score-based generative modeling, Song & Ermon (2019) introduced annealed Langevin dynamics (ALD), which considers a family of smoothed distributions

$$p_t(\boldsymbol{x}) = \int p_{\mathrm{data}}(\boldsymbol{x}_0)\, q_t(\boldsymbol{x} \mid \boldsymbol{x}_0)\,\mathrm{d}\boldsymbol{x}_0, \tag{3}$$

where $q_t(\boldsymbol{x} \mid \boldsymbol{x}_0)$ denotes the corruption kernel and t the diffusion time. Throughout the main development, we use the variance-exploding Gaussian kernel $q_t = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{x}_0, t\boldsymbol{I}_d)$ (Song et al., 2021), for which t is simultaneously the *diffusion time and the noise variance*; we use the two terms interchangeably throughout.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

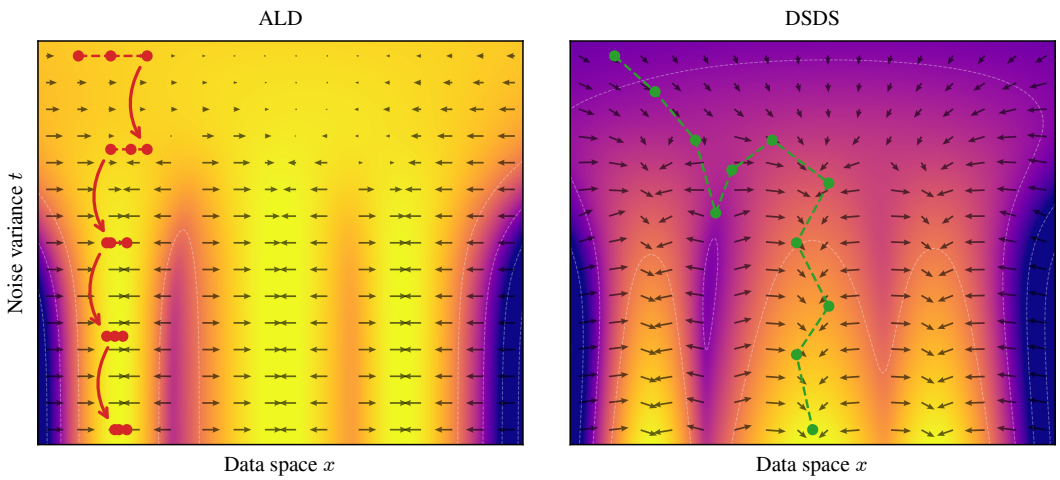


Figure 1: **Conceptual comparison of ALD (left) and DSDS (right).** ALD samples conditionally from $p_t(\boldsymbol{x})$ at fixed noise variance, with visited noise variances given by a predefined schedule. DSDS instead samples jointly from $p(\boldsymbol{x}, t) = p(\boldsymbol{x} \mid t)p(t)$, allowing the noise variance to evolve stochastically across the full space, with $p(t)$ controlling the relative preference over noise variances. Background color indicates density, while arrows illustrate the corresponding score field.

For small t, p_t closely approximates p_{data}, while larger t smooths the density and facilitates transitions between modes. The scores $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$ are learned across noise variances using denoising score matching (Vincent, 2011; Song & Ermon, 2019). At sampling time, ALD applies Langevin updates along a prescribed decreasing schedule $t_1 > \cdots > t_K$. Designing this schedule is therefore an important component of score-based sampling (Karras et al., 2022; Nichol & Dhariwal, 2021), yet the evolution of the noise variance remains externally prescribed.

A related strategy for improving exploration is to augment the sampling state with auxiliary stochastic variables. Tempering methods, for example, introduce an inverse temperature variable that continuously deforms the target as $p_\beta(\boldsymbol{x}) \propto \exp(-\beta U(\boldsymbol{x}))$, thereby facilitating transitions between modes (Marinari & Parisi, 1992; Rissanen et al., 2025). Another prominent example is underdamped Langevin dynamics, which augment the state with a momentum variable $\boldsymbol{p}$ and evolve

$$\mathrm{d}\boldsymbol{x}_s = \boldsymbol{M}^{-1}\boldsymbol{p}_s \mathrm{d}s, \quad \mathrm{d}\boldsymbol{p}_s = \nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x}_s)\mathrm{d}s - \boldsymbol{\Gamma}\boldsymbol{M}^{-1}\boldsymbol{p}_s \mathrm{d}s + \sqrt{2\boldsymbol{\Gamma}}\mathrm{d}\boldsymbol{w}_s, \tag{4}$$

where $\boldsymbol{M}, \boldsymbol{\Gamma} \in \mathbb{R}^{d\times d}$ are symmetric positive-definite mass and friction matrices, respectively. The momentum induces persistent motion, reducing the random-walk behavior of overdamped Langevin dynamics and potentially improving mixing (Cheng et al., 2018; Falk et al., 2025). Related extended-state constructions have also been explored in score-based generative modeling (Dockhorn et al., 2022).

More closely related to our setting, Kim & Ye (2023); Qin & Risteski (2024) promote the time or noise variance itself to a stochastic variable by introducing a prior over noise variances $p(t)$ and consider dynamics in a lifted data-noise space,

$$(\boldsymbol{x}, t) \sim p(\boldsymbol{x}, t) = p(\boldsymbol{x} \mid t)\, p(t).^{1} \tag{5}$$

In both cases, the prior is chosen to encourage excursions to higher noise levels, thereby improving exploration and mixing. However, the resulting joint dynamics are not used to directly generate clean samples. This motivates the central question of this work:

Can a sampling algorithm acting on the joint space $(\boldsymbol{x}, t)$ generate clean samples?

If so, this would alleviate the need to carefully design a prescribed noise schedule that determines which noise levels are visited during sampling (see fig. 1). Instead, one would only specify a prior $p(t)$ encoding the desired overall preference over noise variances, while the actual evolution of t

[1]Since t is now treated as an explicit stochastic variable rather than a prescribed conditioning parameter, we write $p(\boldsymbol{x} \mid t)$ instead of $p_t(\boldsymbol{x})$. See appendix A.1 for details.

would be determined by the joint dynamics. This viewpoint recasts annealed Langevin sampling as a special case of a broader adaptive Langevin dynamics in the joint data-noise space, opening the door to transferring theoretical tools for state-dependent diffusion, including mixing and convergence guarantees.

Our main contributions are:

- We characterize the geometry of the joint space induced by variance-exploding (VE) and variance-preserving (VP) corruption kernels, and use the resulting Fisher information to derive a principled state-dependent Langevin preconditioner.
- We introduce Dual Score Diffusion Sampling (DSDS), a joint-space sampler based on overdamped Langevin dynamics eq. (2), together with a inertial variant (iDSDS) based on underdamped dynamics eq. (4).
- We evaluate DSDS and iDSDS on Gaussian mixture models, MNIST, and CIFAR-10, comparing against ALD and conventional diffusion samplers.

2 Joint Space Characterization

2.1 Constructing the Joint Target

We denote the data space by $\mathcal{X} \subseteq \mathbb{R}^d$ and the noise-variance space by $\mathcal{T} = [t_{\min}, t_{\max}]$. We augment the data variable $\boldsymbol{x} \in \mathcal{X}$ with a stochastic noise variable $t \in \mathcal{T}$ and define the joint state $\boldsymbol{z} = (\boldsymbol{x}, t) \in \mathcal{Z} = \mathcal{X} \times \mathcal{T}$. Given a conditional density $p(\boldsymbol{x} \mid t)$, we introduce a noise prior $p(t)$ and define the joint target using the product rule of probability

$$p(\boldsymbol{z}) = p(\boldsymbol{x}, t) = p(\boldsymbol{x} \mid t)\, p(t). \tag{6}$$

Accordingly, the joint score decomposes as

$$\begin{aligned} \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}, t) &= \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x} \mid t), \\ \partial_t \log p(\boldsymbol{x}, t) &= \partial_t \log p(\boldsymbol{x} \mid t) + \partial_t \log p(t). \end{aligned} \tag{7}$$

If the conditional distribution $p(\boldsymbol{x}, \mid t)$ is not available in closed form, we represent the conditional density by a parametric energy model $U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t)$ (cf. eq. (1)) and learn its spatial and temporal derivatives using dual score matching (Kobler & Pock, 2023; Guth et al., 2025), such that $-\nabla_{\boldsymbol{x}} U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t) \approx \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x} \mid t)$ and $-\partial_t U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t) \approx \partial_t \log p(\boldsymbol{x} \mid t)$. The prior score $\partial_t \log p_{\boldsymbol{\phi}}(t)$ is available analytically in our setting.

Given these quantities, the joint target in eq. (6) can in principle be sampled directly in the augmented space $\mathcal{X} \times \mathcal{T}$ using overdamped Langevin dynamics eq. (2). However, the $\boldsymbol{x}$- and t-directions scale differently with t, particularly as $t \to 0$. In the next section, we characterize this anisotropic geometry and derive a Fisher-informed preconditioner.

2.2 Geometry of the Joint Target

We restrict the main discussion to the VE kernel, while the corresponding VP results are given in appendix B; VE and VP diffusions are related by a change of variables (Karras et al., 2022, Sec. 2).

Score scaling. We begin by investigating how the conditional spatial and variance score components in eq. (7) scale with t. In contrast, the scaling of the prior score is fully determined by the choice of prior and is therefore designable.

Proposition 2.1 (Score scaling)**.** *Under the VE corruption process, the conditional spatial and temporal scores obey, for every $t > 0$ and $i = 1, \dots, d$,*

$$\mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x} \mid t)} \left[\left(\partial_{x_i} \log p(\boldsymbol{x} \mid t) \right)^2 \right] \leq \frac{1}{t} \quad \textit{and} \quad \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x} \mid t)} \left[\left(\partial_t \log p(\boldsymbol{x} \mid t) \right)^2 \right] \leq \frac{d}{2t^2}. \tag{8}$$

The result follows by expressing the marginal scores as conditional expectations of the corresponding Gaussian kernel scores and applying Jensen's inequality. The proof is given in appendix A.2.

Algorithm 1: ALD

Input: Noise schedule $t_0 > \cdots > t_K$
State $\boldsymbol{x}$
Conditional score $s_t(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$
Preconditioner $\boldsymbol{A}_t = t\boldsymbol{I}$
Initialize: $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, t_0 \boldsymbol{I}_d)$
for $k = 0$ **to** K **do**
  $t \leftarrow t_k$
  for $\ell = 1$ **to** L **do**
    $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$
    $\boldsymbol{x} \leftarrow \boldsymbol{x} + \eta\, \boldsymbol{A}_t s_t(\boldsymbol{x}) + \sqrt{2\eta \boldsymbol{A}_t}\boldsymbol{\xi}$
return $\boldsymbol{x}$

Algorithm 2: DSDS

Input: Noise-variance prior $p(t)$, initial t_0
State $\boldsymbol{z} = (\boldsymbol{x}, t)$
Joint score $s(\boldsymbol{x}, t) = \nabla_{(\boldsymbol{x},t)} \log(p(\boldsymbol{x} \mid t)p(t))$
Preconditioner $\boldsymbol{A}(t) = \mathrm{diag}(t\boldsymbol{I}, a(t))$
Initialize: $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, t_0 \boldsymbol{I}_d)$, $\boldsymbol{z} \leftarrow (\boldsymbol{x}, t_0)$
for $\ell = 1$ **to** N **do**
  $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{d+1})$
  $\begin{pmatrix}\boldsymbol{x}\\ t\end{pmatrix} \leftarrow \begin{pmatrix}\boldsymbol{x}\\ t\end{pmatrix} + \eta\big[\boldsymbol{A}(t)s(\boldsymbol{x},t) + \nabla\cdot\boldsymbol{A}(t)\big] + \sqrt{2\eta \boldsymbol{A}(t)}\boldsymbol{\xi}$
  $t \leftarrow \mathrm{Reflect}_{[t_{\min}, t_{\max}]}(t)$
return $\boldsymbol{x}$

Fisher-informed preconditioning. To account for the scale mismatch, we use the Fisher information to construct a local preconditioner, following Titsias (2023). Since the Fisher information associated with $p(\boldsymbol{x} \mid t)$, and consequently with the joint target $p(\boldsymbol{x}, t)$, is generally intractable, we instead use that induced by the VE kernel and variance prior, $q(\boldsymbol{x}, t \mid \boldsymbol{x}_0) = q(\boldsymbol{x} \mid \boldsymbol{x}_0, t), p(t)$.

Proposition 2.2 (VE-induced Fisher metric)**.** *For the joint variable $\boldsymbol{z} = (\boldsymbol{x}, t)$, the Fisher metric induced by the VE kernel and variance prior $p(t)$ is*

$$\boldsymbol{\mathcal{I}}(t) = \underbrace{\begin{pmatrix} t^{-1}\boldsymbol{I}_d & \boldsymbol{0} \\ \boldsymbol{0}^\top & \frac{d}{2t^2} \end{pmatrix}}_{\textit{conditional}} + \underbrace{\begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0}^\top & (\partial_t \log p(t))^2 \end{pmatrix}}_{\textit{prior}}. \tag{9}$$

The metric is obtained by substituting the closed-form VE kernel into the definition of the Fisher information; the full derivation is given in appendix A.3. Following Titsias (2023), we define the preconditioner as its inverse, $\boldsymbol{A}(t) = \boldsymbol{\mathcal{I}}(t)^{-1}$.

Log-variance reparameterization. The conditional contribution to the variance Fisher component $\mathcal{I}_{tt}$ in eq. (9) scales as t^{-2}, leading to increasing anisotropy as $t \to 0$. We therefore reparameterize the noise variance coordinate as $\tau = \log t$, with $\tau \in \mathcal{T}_\tau = [\tau_{\min}, \tau_{\max}]$, and, with slight abuse of notation, write the joint state as $\boldsymbol{z} = (\boldsymbol{x}, \tau)$. This transformation requires no model retraining, since the conditional and prior scores transform analytically as

$$\partial_\tau \log p(\boldsymbol{x} \mid \tau) = t\, \partial_t \log p(\boldsymbol{x} \mid t), \qquad \partial_\tau \log p_\tau(\tau) = 1 + t\, \partial_t \log p(t), \tag{10}$$

where the transformed prior density is given by $p_\tau(\tau) = p(t)\, t$. The log transformation removes this explicit t^{-2} scaling from the conditional contribution to $\mathcal{I}_{\tau\tau}$.

Proposition 2.3 (Fisher metric in log-variance coordinates)**.** *In coordinates $\boldsymbol{z} = (\boldsymbol{x}, \tau)$, the Fisher metric induced by the VE kernel and transformed variance prior $p_\tau(\tau)$ is*

$$\boldsymbol{\mathcal{I}}(\tau) = \begin{pmatrix} e^{-\tau}\boldsymbol{I}_d & \boldsymbol{0} \\ \boldsymbol{0}^\top & \dfrac{d}{2} + (\partial_\tau \log p_\tau(\tau))^2 \end{pmatrix}. \tag{11}$$

The result follows directly from the change of variables; the derivation is given in appendix A.4.

Choice of variance prior. After the log-variance reparameterization, the remaining τ-dependence of $\mathcal{I}_{\tau\tau}$ enters only through the log prior score. For a Pareto prior $p(t) \propto t^{-(\alpha+1)}$, $\alpha > 0$, we have $\partial_\tau \log p_\tau(\tau) = -\alpha$, so that $\mathcal{I}_{\tau\tau} = \frac{d}{2} + \alpha^2$ (and thus the variance preconditioner) is constant. Alternative priors are studied in section 5.2.

3 Dual Score Diffusion Sampling

3.1 Overdamped DSDS

We construct sampling dynamics for the augmented state $\boldsymbol{z} = (\boldsymbol{x}, t)$ using preconditioned overdamped Langevin dynamics. State dependence of the preconditioner $\boldsymbol{A}(\boldsymbol{z})$ requires a divergence

correction to preserve the target density (Xifara et al., 2014, Eq. (9)), while the bounded time domain $\mathcal{T}$ is enforced by reflection (Lou & Ermon, 2023, Eqs. (40),(41)). This gives the reflected preconditioned Langevin dynamics

$$\mathrm{d}\boldsymbol{z}_s = [\boldsymbol{A}(\boldsymbol{z}_s)\nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}_s) + \nabla_{\boldsymbol{z}} \cdot \boldsymbol{A}(\boldsymbol{z}_s)]\,\mathrm{d}s + \sqrt{2\boldsymbol{A}(\boldsymbol{z}_s)}\mathrm{d}\boldsymbol{w}_s + \boldsymbol{A}(\boldsymbol{z}_s)\boldsymbol{n}(\boldsymbol{z}_s)\,\mathrm{d}L_s, \tag{12}$$

where L_s denotes the boundary local time, $\boldsymbol{n}(\boldsymbol{z})$ the inward unit normal to the boundary, and $\boldsymbol{A}(\boldsymbol{z})\boldsymbol{n}(\boldsymbol{z})$ the corresponding conormal reflection direction. The matrix divergence is defined componentwise by $(\nabla \cdot \boldsymbol{A})_i = \sum_j \partial_{z_j} A_{ij}$. A derivation of eq. (12) is given in appendix C.1.

Full algorithm. For the actual sampler, we express eq. (12) in the log-variance coordinate τ, specialize it to the Fisher-informed preconditioner from section 2.2, and discretize the resulting dynamics with Euler-Maruyama using step size η. For this preconditioner, the divergence correction is available in closed form; see appendix C.1.3. After each step, the proposed log-variance coordinate is mirror-reflected into $\mathcal{T}_\tau$. While the Fisher preconditioner accounts for the noise-variance anisotropy, the conditional and prior scores exhibit different scaling with the data dimension d. We therefore apply an additional dimensionality rescaling; details are given in appendix C.1.4. Algorithm 2 lists the high-level structure of the proposed DSDS sampler, while the full sampler is detailed in algorithm 3.

3.2 Accelerating DSDS with Momentum

To accelerate exploration, we augment DSDS with momentum by applying underdamped Langevin dynamics eq. (4) in the joint space $(\boldsymbol{x}, t)$, with momentum $\boldsymbol{p} = (\boldsymbol{p}_{\boldsymbol{x}}, p_t)$. Following the Riemannian construction of Girolami & Calderhead (2011, Eq. (15)), we incorporate the Fisher-informed preconditioner through the state-dependent mass matrix $\boldsymbol{M}(\boldsymbol{z}) = \boldsymbol{A}(\boldsymbol{z})^{-1}$. Its state dependence introduces the metric correction $\boldsymbol{b}_{\boldsymbol{A}}(\boldsymbol{z}, \boldsymbol{p}) = \frac{1}{2}\nabla_{\boldsymbol{z}} \log\det \boldsymbol{A}(\boldsymbol{z}) - \frac{1}{2}\nabla_{\boldsymbol{z}}\big(\boldsymbol{p}^\top \boldsymbol{A}(\boldsymbol{z})\boldsymbol{p}\big)$. At the boundary of the time domain, we impose specular reflection, the kinetic counterpart of reflecting overdamped dynamics (Du et al., 2026, Eq. (4)). The resulting dynamics are

$$\begin{aligned}
\mathrm{d}\boldsymbol{z}_s &= \boldsymbol{A}(\boldsymbol{z}_s)\boldsymbol{p}_s\,\mathrm{d}s, \\
\mathrm{d}\boldsymbol{p}_s &= \big[\nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}_s) + \boldsymbol{b}_{\boldsymbol{A}}(\boldsymbol{z}_s, \boldsymbol{p}_s) - \boldsymbol{\Gamma}(\boldsymbol{z}_s)\boldsymbol{A}(\boldsymbol{z}_s)\boldsymbol{p}_s\big]\mathrm{d}s + \sqrt{2\boldsymbol{\Gamma}(\boldsymbol{z}_s)}\mathrm{d}\boldsymbol{w}_s, \qquad \boldsymbol{z}_s \in \mathcal{Z}, \\
\boldsymbol{p}_{s^+} &= \mathcal{R}_{\boldsymbol{z}_s}(\boldsymbol{p}_{s^-}), \qquad \boldsymbol{z}_s \in \partial\mathcal{Z}.
\end{aligned} \tag{13}$$

Here, $\boldsymbol{\Gamma}(\boldsymbol{z})$ denotes the friction matrix, $\boldsymbol{p}_{s^-}$ and $\boldsymbol{p}_{s^+}$ the momenta immediately before and after reflection, respectively, and $\mathcal{R}_{\boldsymbol{z}}$ the corresponding specular reflection map. The derivation is given in appendix C.2.

Full algorithm. For the inertial DSDS (short iDSDS), we set $\boldsymbol{\Gamma}(\boldsymbol{z}) = \varepsilon \boldsymbol{A}(\boldsymbol{z})^{-1}$, which reduces the friction term to the uniform damping $-\varepsilon\boldsymbol{p}$ following Falk et al. (2025). We then apply the log-variance reparameterization and discretize the dynamics using their inertial Langevin scheme, yielding a state-only update in the joint $(\boldsymbol{x}, \tau)$ space. The log-variance component is specularly reflected at the boundaries of $\mathcal{T}_\tau$, and the same dimensionality rescaling as for DSDS is applied. The resulting sampler is summarized in algorithm 4.

3.3 Sample Readout

For both proposed joint approaches, we discard the time coordinate and return $\boldsymbol{x}$ after the final update. At stationarity, the resulting marginal satisfies the following bound.

Proposition 3.1 (Sample accuracy)**.** *Suppose the joint sampler follows its invariant distribution $p(\boldsymbol{x}, t) = p(\boldsymbol{x} \mid t)p(t)$, where $p(\boldsymbol{x} \mid t)$ is the VE-corrupted data distribution. Then the marginal $p_{\boldsymbol{x}}(\boldsymbol{x}) = \int_{t_{\min}}^{t_{\max}} p(\boldsymbol{x} \mid t)\, p(t)\,\mathrm{d}t$ satisfies*

$$\mathcal{W}_2(p_{\boldsymbol{x}}, p_{\mathrm{data}}) \le \sqrt{d\,\mathbb{E}_{t\sim p(t)}[t]}. \tag{14}$$

The bound follows from constructing an explicit coupling between clean and VE-corrupted samples; see appendix C.3. Note that it captures only the smoothing error, excluding model approximation,

finite-time, and discretization effects. For comparison, exact sampling from the terminal ALD distribution gives $\mathcal{W}_2(p_{t_K}, p_{\mathrm{data}}) \leq \sqrt{d\, t_K}$. Thus, in this bound, DSDS replaces the terminal noise level t_K by the prior mean $\mathbb{E}_{p(t)}[t]$, exposing a trade-off between low-noise sample accuracy and high-noise exploration.

4 Related work

Prior work on joint $(\boldsymbol{x}, t)$ space. Kim & Ye (2023) use joint-space MCMC as a warm start for a subsequent reverse-SDE/ODE sampler. Their DLG method combines overdamped dynamics in $\boldsymbol{x}$ with Gibbs updates in t, avoiding a learned time score. In contrast, DSDS uses state-dependent preconditioned Langevin dynamics in both coordinates and generates the final sample directly from the joint process. Qin & Risteski (2024) study joint $(\boldsymbol{x}, t)$-space overdamped Langevin dynamics only theoretically, focusing on mode exploration and mixing rather than sample generation, and without any preconditioning.

Training- and schedule-based approaches. Our method requires learning an additional time score. Related approaches learn components of the diffusion process, including the noise schedule (Kingma et al., 2021; San-Roman et al., 2021; Sahoo et al., 2024), or optimize the sampling schedule of a pretrained model (Sabour et al., 2024; Chen et al., 2024). In both cases, the noise variances visited during sampling remain prescribed rather than evolving as part of the sampling.

5 Experiments

We organize the experiments into two parts. First, we isolate sampling behavior using GMMs with analytical ground-truth scores, allowing us to study the sampler independently of model error; second, we evaluate the learned energy model on image data and examine how the proposed parameterization scales to higher dimensional settings.

Hyperparameter. We provide recommended default values for the main DSDS hyperparameters $\eta = 0.5$ and $\alpha = 4$ (and $\beta = 0.8$ for iDSDS), while tuning all sampler-specific hyperparameters, including those of the baselines, separately for each experimental setting using Optuna (Akiba et al., 2019).

5.1 Low-dimensional Gaussian Mixture Models

Experimental details. As baseline, we include the unadjusted Langevin algorithm (ULA), the recent inertial Langevin algorithm (ILA) (Falk et al., 2025), and annealed Langevin dynamics (ALD) (Song & Ermon, 2019). For the numerical evaluation, we report the 2-Wasserstein distance defined as $W_2(p, p_{\mathrm{data}}) = \left(\inf_{\pi \in \Pi(p, p_{\mathrm{data}})} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\pi}[\|\boldsymbol{x}-\boldsymbol{y}\|_2^2]\right)^{1/2}$ for $d = 1$, and its sliced variant with $N_{\mathrm{slices}} = 1000$ random projections for $d > 1$, using $N_{\mathrm{samples}} = 50\,000$ samples.

Sampling. We first evaluate sampling performance on the 1D and 2D Gaussian mixture models introduced in appendix D.1; results are shown in fig. 2 (left, middle). In 1D, we initialize $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, t_{\max}\boldsymbol{I}_d)$, whereas in 2D we follow Falk et al. (2025) and use $\boldsymbol{x}_0 \sim \mathcal{N}((1.5, 1.5)^\top, 0.1^2\boldsymbol{I}_2)$. Despite sampling jointly over data and noise level, DSDS matches the convergence of the strong ALD baseline, while iDSDS exploits the joint dynamics to reach low W_2 substantially faster. The advantage becomes particularly apparent in the 2D mixture, where ULA and ILA plateau at considerably higher error for the considered number of update steps, whereas both DSDS variants converge to the low-error regime. The corresponding noise-level trajectories in fig. 2 (right) illustrate how DSDS dynamically traverses the noise dimension during sampling.

5.2 Ablation Study

We conduct a series of ablations on the 1D Gaussian mixture model (GMM) problem to isolate the contribution of DSDS's individual design choices: the initial noise variance t_0, the choice of prior distribution over t, the parameterization of the noise-variance, and the preconditioning. For each

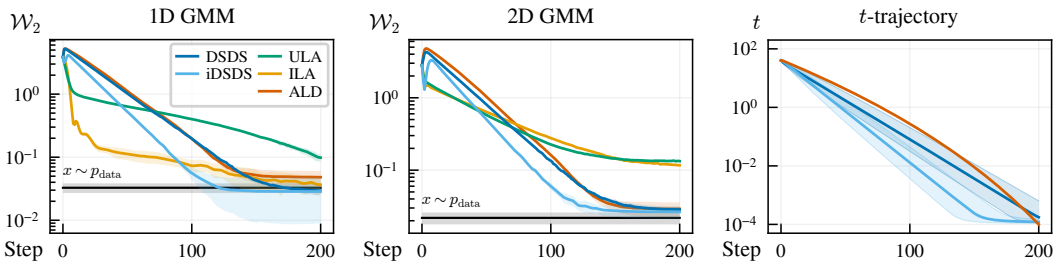


Figure 2: GMM sampling results.

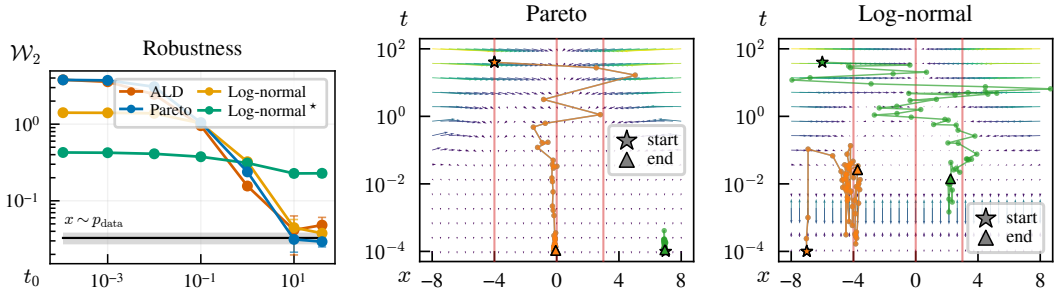


Figure 3: Scaling and robustness experiments.

configuration, all method-specific hyperparameters (e.g., step size η, prior parameters) were tuned individually for each budget of update steps N.

Robustness to initial noise-variance choice. We assess the robustness of DSDS to the initial noise variance t_0 by repeating the 1D GMM experiment with t_0 varied from $t_{\max}$ to $t_{\min}$ and reporting the W_2 distance after 200 sampling steps; ALD is included as a reference fig. 3 (left). The results reveal a trade-off between fast sampling and robustness to initialization. Priors optimized for rapid sampling from $t_{\max}$ (Pareto - blue curve; log-normal - yellow curve) exhibit a stronger dependence on t_0, similar to ALD, whereas the robustness-oriented log-normal* (green curve) maintains nearly constant performance across the full range of initial noise levels, at the cost of slower convergence when initialized at $t_{\max}$. The corresponding drift fields and representative trajectories in fig. 3 (middle, right) illustrate how the choice of noise-variance prior changes the joint dynamics, demonstrating that DSDS can be configured either for fast sampling or for increased robustness to initialization.

Prior distribution. We compare Pareto, log-normal, and power-exponential priors. In log-variance τ, their prior scores are respectively constant $(-\alpha)$, linear $(-(\tau - \mu)/\sigma^2)$, and nonlinear $(1 - \lambda q e^{q\tau})$. More complex τ-dependent prior scores do not improve fast sampling over the constant-score Pareto prior (fig. 4, right). We therefore use Pareto throughout. The exact definition of each prior and additional priors are included in appendix E.3.

Variance parametrization. We compare Langevin updates in normal noise-variance t, in log variance τ, and a Gibbs update over τ inspired by Kim & Ye (2023) (see appendix D.4.3). The log-variance reparameterization improves sampling efficiency with a slight speed-up over the normal noise-variance variable particularly for a lower number of total sampling steps N. In contrast, the Gibbs update converges slower and appears less effective for generating high-quality samples, see fig. 4, middle.

Preconditioning. We compare no preconditioning (Euclidean $\boldsymbol{A} = \boldsymbol{I}_{d+1}$), global Fisher preconditioning, and local curvature-aware preconditioning for DSDS. The curvature preconditioner is constructed from the Eigen-decomposition of the joint potential Hessian $\nabla^2_{\boldsymbol{z}} \log p(\boldsymbol{x}, t)$ following Falk et al. (2026); see appendix A.5 for the closed form derivation. Fisher preconditioning performs best

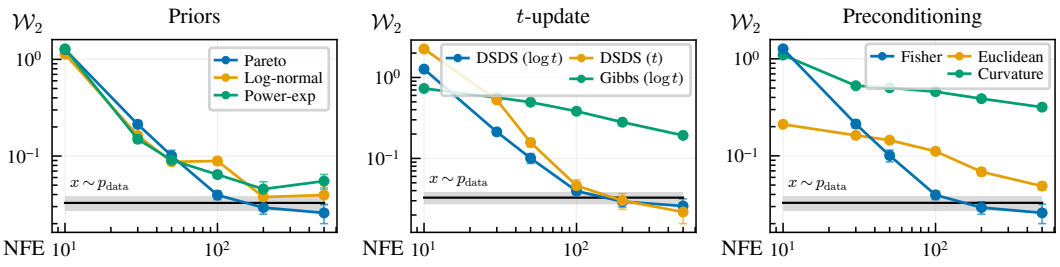
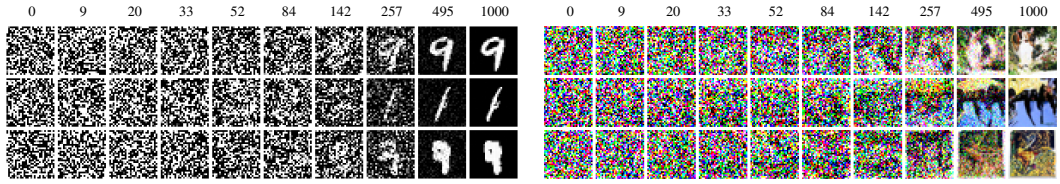
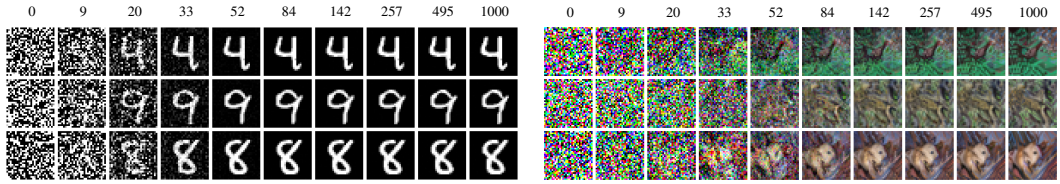


Figure 4: DSDS ablation study.



(a) DSDS



(b) iDSDS

Figure 5: Overview of intermediate MNIST (left) and CIFAR-10 (right) samples. Column titles indicate the respective step.

while adding negligible overhead due to its closed-form structure, whereas curvature-aware preconditioning is substantially more expensive and converges more slowly. The faster initial progress of the Euclidean variant results from a stronger effective drift toward small t; we analyze this effect and the Fisher preconditioner in more detail in appendix E.4.

5.3 Image Generation

Experimental details. We parameterize the energy with a modified NCSN++ U-Net (Song et al., 2021) based on the EDM implementation (Karras et al., 2022). The backbone produces a feature representation that is mapped to the scalar energy $U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t)$ by the energy head; details see appendix D.2. As baselines, we include ALD, the Euler-Maruyama discretization of the reverse SDE (EM), and its predictor-corrector (PC) variant (Song et al., 2021). We report the Fréchet Inception Distance (FID) (Heusel et al., 2017) and the negative log-likelihood (NLL) $\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[U_{\boldsymbol{\theta}}(\boldsymbol{x}, t_{\min})]$, which is available from the energy model trained with dual score matching (Guth et al., 2025). Note that a lower NLL indicates that the sampler generates data which better fits the model's distribution, even if the model did not learn the data distribution well. We generate $N_{\text{samples}} = 10\,000$ samples.

Results. On MNIST (table 1a), iDSDS clearly outperforms ALD across all budgets, leading in both FID and NLL at NFE 100/500 and remaining ahead at NFE 1000, while plain DSDS starts far behind ALD at low NFE and only surpasses it by NFE 1000. On CIFAR-10 (table 1b), iDSDS again outperforms ALD at every budget by a wide margin in both FID and NLL, though its FID worsens slightly from NFE 1000 to 2000 despite a continued NLL decrease; DSDS trails ALD substantially at low NFE but closes the gap by NFE 2000. Overall, both DSDS variants – and iDSDS in particular – consistently improve on the annealing-based ALD baseline, narrowing much of the gap to the reverse SDE-based EM and PC samplers, with remaining headroom at low NFE for DSDS and high NFE for iDSDS on CIFAR-10 pointing to promising directions for further gains.

Table 1: Image sampling performance at representative NFE budgets. FID and NLL are computed from 10,000 generated samples. NLL is reported in bits per dimension. Entries show mean $\pm$ standard deviation over $\{3, 5\}$ different seeds for CIFAR-10 and MNIST repsectively. Best results are bolded and second-best results are underlined. Full NFE sweeps are reported in the appendix.

(a) MNIST.

	NFE 100		NFE 500		NFE 1000	
Method	FID ↓	NLL ↓	FID ↓	NLL ↓	FID ↓	NLL ↓
ALD	58.35 ± 0.79	3.13 ± 0.33	18.85 ± 0.20	2.49 ± 0.23	15.31 ± 0.33	2.37 ± 0.21
EM	13.38 ± 0.24	$\underline{2.53} \pm 0.23$	9.98 ± 0.11	2.31 ± 0.21	$\underline{9.79} \pm 0.09$	2.29 ± 0.23
PC	$\underline{11.86} \pm 0.10$	2.55 ± 0.22	$\mathbf{7.10} \pm 0.09$	$\underline{2.27} \pm 0.20$	$\mathbf{6.74} \pm 0.14$	2.22 ± 0.21
DSDS	310.18 ± 0.44	$7.3\times10^{3} \pm 4.2\times10^{4}$	45.19 ± 0.12	$10.45 \pm 3.3\times10^{2}$	14.68 ± 0.18	$\mathbf{2.20} \pm 0.17$
iDSDS	$\mathbf{9.36} \pm 0.14$	$\mathbf{2.40} \pm 0.21$	$\underline{8.99} \pm 0.15$	$\mathbf{2.24} \pm 0.18$	11.05 ± 0.08	$\underline{2.21} \pm 0.16$

(b) CIFAR-10.

	NFE 500		NFE 1000		NFE 2000	
Method	FID ↓	NLL ↓	FID ↓	NLL ↓	FID ↓	NLL ↓
ALD	144.43 ± 0.89	5.99 ± 0.46	101.37 ± 0.61	5.46 ± 0.73	67.72 ± 0.20	4.68 ± 0.86
EM	$\underline{46.68} \pm 0.28$	$\mathbf{3.62} \pm 0.54$	$\underline{42.99} \pm 0.23$	$\mathbf{3.45} \pm 0.52$	$\underline{41.85} \pm 0.52$	$\mathbf{3.39} \pm 0.51$
PC	$\mathbf{38.40} \pm 0.42$	$\underline{3.89} \pm 0.58$	$\mathbf{31.22} \pm 0.24$	$\underline{3.70} \pm 0.53$	$\mathbf{28.58} \pm 0.31$	3.50 ± 0.55
DSDS	341.16 ± 0.14	8.90 ± 0.33	98.20 ± 0.70	5.62 ± 0.48	68.67 ± 0.77	4.89 ± 0.63
iDSDS	82.43 ± 0.38	4.19 ± 0.56	81.98 ± 0.45	4.20 ± 0.47	94.05 ± 0.82	$\underline{3.39} \pm 0.39$

6 Conclusion

We introduced Dual Score Diffusion Sampling (DSDS), which lifts annealed Langevin sampling to the joint data-variance space $(\boldsymbol{x}, t)$ and lets the noise-variance coordinate evolve stochastically rather than following a prescribed schedule. By characterizing the geometry induced by the corruption kernel, we derived a Fisher-informed preconditioner that accounts for the anisotropic scaling between the spatial and noise-variance directions, and we showed that a simple log-variance reparameterization together with a Pareto prior renders this preconditioner constant. Building on this geometry, we proposed both an overdamped variant (DSDS) and a inertial variant (iDSDS). On Gaussian mixture models, iDSDS in particular converges substantially faster than strong schedule-based and inertial baselines, and our ablations show that DSDS is robust to the choice of initial noise variance, unlike ALD, provided the prior is chosen accordingly. On imaging data, iDSDS is competitive with, and at low NFE budgets can exceed, ALD and reverse-SDE samplers, demonstrating that clean samples can indeed be generated directly from joint-space dynamics without an externally imposed schedule.

7 Limitations

Our theoretical guarantee (proposition 3.1) only bounds the asymptotic smoothing error at stationarity and excludes score approximation, finite-time, and discretization errors, which likely dominate in practice, particularly at low NFE. Moreover, the Fisher preconditioner is derived from the corruption kernel as a tractable proxy for the intractable joint distribution.

Further empirical evaluation is also needed. Plain DSDS performs poorly at low NFE on MNIST and CIFAR-10 and only becomes competitive at larger budgets, potentially due to limited gradient accuracy of the learned joint distribution, whereas iDSDS is more robust at small budgets. We have only evaluated the Fisher preconditioner on low-dimensional GMMs and two standard image benchmarks; its effectiveness for other corruption processes, higher-resolution data, and architectures remains open. Finally, DSDS requires learning an additional time score, increasing training cost and introducing an additional source of approximation error compared with schedule-based samplers.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods and assisting in the writing of proofs. We did not use generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, or interpret results. Generating synthetic datasets, assisting with translation, cleaning and reformatting datasets, and supporting qualitative or thematic data analysis were not applicable to this work.

Additionally, we used generative AI tools for sourcing and searching for information, identifying relevant literature, creating and editing software code, and editing the manuscript to improve readability.

We reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by the authors. Claims concerning existing literature were verified by consulting the relevant parts of the cited publications. For mathematical content, generative AI tools were used to assist with the presentation of proofs, clean up notation, and cross-check derivations; the mathematical claims and critical arguments underlying their proofs were developed and verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We do not identify additional ethical concerns specific to this work beyond those generally associated with generative modeling. Our method is a general-purpose sampling framework evaluated on standard benchmark datasets and does not introduce application-specific risks.

REPRODUCIBILITY STATEMENT

Theoretical results: For each proposition/theorem, we state the proof idea in the main text and provide the complete proof (see appendices A to C).
Numerical results: We provide detailed descriptions of all datasets, model architectures, and algorithms used in the appendix (see appendix D). We further report all random seeds used in our experiments.

AUTHOR CONTRIBUTIONS

Author contributions are omitted for anonymous review and will be included in the camera-ready version.

ACKNOWLEDGMENTS

Acknowledgments are omitted for anonymous review and will be included in the camera-ready version.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: a next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631. ACM, 2019. doi: 10.1145/3292500.3330701.

Brian D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, May 1982. ISSN 0304-4149. doi: 10.1016/0304-4149(82)90051-5.

Yuzhu Chen, Fengxiang He, Shi Fu, Xinmei Tian, and Dacheng Tao. Adaptive Time-Stepping Schedules for Diffusion Models. In *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, pp. 685–697. PMLR, September 2024.

Xiang Cheng, Niladri S. Chatterji, Peter L. Bartlett, and Michael I. Jordan. Underdamped Langevin MCMC: A non-asymptotic analysis. In *Proceedings of the 31st Conference On Learning Theory*, pp. 300–323. PMLR, July 2018.

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped Langevin diffusion. In *International Conference on Learning Representations*, 2022.

Hengrong Du, Qi Feng, and Lingjiong Zhu. Sharp hypocoercive convergence estimates for underdamped Langevin dynamics with specular reflection, August 2026. arXiv:2608.17022 [math.PR].

Alexander Falk, Andreas Habring, Christoph Griesbacher, and Thomas Pock. An Inertial Langevin Algorithm, October 2025. arXiv:2510.06723 [math.NA].

Alexander Falk, Laurenz Nagler, Andreas Habring, and Thomas Pock. Time-inhomogeneous preconditioned Langevin dynamics, 2026. arXiv:2605.06091.

Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2): 123–214, March 2011. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2010.00765.x.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, Cambridge, MA, USA, 2016. ISBN 978-0-262-03561-3.

Florentin Guth, Zahra Kadkhodaie, and Eero P. Simoncelli. Learning normalized image densities via dual score matching. In *Advances in Neural Information Processing Systems*, volume 38, pp. 89796–89826, 2025. doi: 10.52202/085713-3002.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 978-1-5108-6096-4.

Jonathan Ho, Ajay N. Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577, 2022. doi: 10.52202/068431-1926.

Beomsu Kim and Jong Chul Ye. Denoising MCMC for accelerating diffusion-based generative models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of machine learning research*, pp. 16955–16977. Proceedings of Machine Learning Research, 2023.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational Diffusion Models. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21696–21707. Curran Associates, Inc., 2021.

Erich Kobler and Thomas Pock. Learning gradually non-convex image priors using score matching, 2023. arXiv:2302.10502.

Erich Kobler, Alexander Effland, Karl Kunisch, and Thomas Pock. Total deep variation for linear inverse problems. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7546–7555. IEEE, 2020.

Erich Kobler, Alexander Effland, Karl Kunisch, and Thomas Pock. Total deep variation: A stable regularization method for inverse problems. *IEEE transactions on pattern analysis and machine intelligence*, 44(12):9163–9180, 2021. URL `https://ieeexplore.ieee.org/abstract/document/9599362/`.

A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical Report, University of Toronto, 2009.

Y. Le Cun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. ISSN 1558-2256. doi: 10.1109/5.726791.

Aaron Lou and Stefano Ermon. Reflected diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of machine learning research*, pp. 22675–22701. Proceedings of Machine Learning Research, 2023.

Enzo Marinari and Giorgio Parisi. Simulated tempering: a new Monte Carlo scheme. *Europhysics Letters*, 19(6):451–458, 1992. doi: 10.1209/0295-5075/19/6/002.

James Martin, Lucas C. Wilcox, Carsten Burstedde, and Omar Ghattas. A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion. *SIAM Journal on Scientific Computing*, 34(3):A1460–A1487, 2012. doi: 10.1137/110845598.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of machine learning research*, pp. 8162–8171. Proceedings of Machine Learning Research, 2021.

Yilong Qin and Andrej Risteski. Fit like you sample: Sample-efficient generalized score matching from fast mixing diffusions. In *Proceedings of the thirty seventh Conference on Learning Theory*, volume 247 of *Proceedings of machine learning research*, pp. 4413–4457. Proceedings of Machine Learning Research, 2024.

Severi Rissanen, Ruikang Ouyang, Jiajun He, Wenlin Chen, Markus Heinonen, Arno Solin, and José Miguel Hernández-Lobato. Progressive tempering sampler with diffusion. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of machine learning research*, pp. 51724–51746. Proceedings of Machine Learning Research, 2025.

Gareth O. Roberts and Richard L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996. doi: 10.2307/3318418.

Stefan Roth and Michael J. Black. Fields of Experts. *International Journal of Computer Vision*, 82 (2):205–229, April 2009. ISSN 1573-1405. doi: 10.1007/s11263-008-0197-6.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your steps: optimizing sampling schedules in diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pp. 42947–42975, Vienna, Austria, July 2024. JMLR.org.

Subham Sekhar Sahoo, Aaron Gokaslan, Christopher De Sa, and Volodymyr Kuleshov. Diffusion Models With Learned Adaptive Noise. In *Advances in Neural Information Processing Systems*, volume 37, pp. 105730–105779, November 2024. doi: 10.52202/079017-3354.

Robin San-Roman, Eliya Nachmani, and Lior Wolf. Noise Estimation for Generative Diffusion Models, 2021. arXiv:2104.02600 [cs.LG].

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, volume 32, pp. 11895–11907, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

A. M. Stuart. Inverse problems: a Bayesian perspective. *Acta Numerica*, 19:451–559, 2010. doi: 10.1017/S0962492910000061.

Yee Whye Teh, Alexandre H. Thiéry, and Sebastian J. Vollmer. Consistency and fluctuations for stochastic gradient Langevin dynamics. *Journal of Machine Learning Research*, 17(7):1–33, 2016.

Michalis K. Titsias. Optimal preconditioning and Fisher adaptive Langevin sampling. In *Advances in Neural Information Processing Systems*, volume 36, pp. 29449–29460, 2023. doi: 10.52202/075280-1280.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 681–688, 2011.

Tatiana Xifara, Chris Sherlock, Samuel Livingstone, Simon Byrne, and Mark Girolami. Langevin diffusions and the Metropolis-adjusted Langevin algorithm. *Statistics & Probability Letters*, 91: 14–19, 2014. doi: 10.1016/j.spl.2014.04.002.

Table 2: Notation used throughout the paper.

Symbol	Description
$d \in \mathbb{N}$	Dimension of data samples
$\boldsymbol{x}_0 \in \mathbb{R}^d$	Clean data sample with density $p_{\mathrm{data}}(\boldsymbol{x}_0)$
$t \in [t_{\min}, t_{\max}]$	Noise variance; synonym for diffusion time.
$\boldsymbol{x} \in \mathbb{R}^d$	Corrupted data sample with density $p(\boldsymbol{x} \mid t)$
$q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)$	Corruption kernel conditioned on the clean sample $\boldsymbol{x}_0$. The main body considers the VE kernel eq. (15); additional calculations for the VP kernel eq. (53) are given in appendix B.
$p(\boldsymbol{x} \mid t)$	Corrupted data marginal at noise variance t, obtained by integrating $q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)$ over $p_{\mathrm{data}}(\boldsymbol{x}_0)$.
$p_t(\boldsymbol{x})$	Shorthand for $p(\boldsymbol{x} \mid t)$ when t is treated as a fixed parameter, as in ALD.
$\tau = \log t$	Log-variance coordinate
$p(t)$	Variance prior in normal variance space t
$p_\tau(\tau)$	Variance prior in log variance space τ
$p(\boldsymbol{x} \mid \tau)$	Conditional distribution in log variance (with slight abuse of notation writing $p(\boldsymbol{x} \mid \tau)$ instead of $p(\boldsymbol{x} \mid e^\tau)$.
$\boldsymbol{z} = (\boldsymbol{x}, \tau)$	Joint state in data-log-variance space
$p(\boldsymbol{x}, t)$	Joint distribution $p(\boldsymbol{x}, t) = p(\boldsymbol{x} \mid t)p(t)$ in data-variance space
$p(\boldsymbol{x}, \tau)$	Joint distribution $p(\boldsymbol{x}, \tau) = p(\boldsymbol{x} \mid \tau)p_\tau(\tau)$ in data-log-variance space
s	Langevin sampling time
$U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t)$	Energy model parameterizing $p(\boldsymbol{x} \mid t)$ following eq. (1)

A Geometry Characterization for the VE Kernel

A.1 Notation and Definitions

We follow the notation of Goodfellow et al. (2016) as implemented in the accompanying `math_commands.tex` file. Vectors are denoted by bold lowercase letters and matrices by bold uppercase letters. We use q for distributions conditioned on the clean variable $\boldsymbol{x}_0$ and p for the corresponding marginals after $\boldsymbol{x}_0$ has been integrated out. An overview of the objects used throughout the paper is provided in table 2.

A.1.1 Preliminaries

Throughout the proofs, we consider the Gaussian VE corruption kernel

$$q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) \coloneqq \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{x}_0, t\, \boldsymbol{I}_d) = \frac{1}{(2\pi t)^{d/2}} \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_0\|^2}{2t}\right), \tag{15}$$

which can equivalently be defined via

$$\boldsymbol{x} = \boldsymbol{x}_0 + \sqrt{t}\, \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d). \tag{16}$$

The conditional spatial and temporal scores of the corruption kernel are

$$\begin{aligned} \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) &= -\frac{\boldsymbol{x} - \boldsymbol{x}_0}{t} = -\frac{\boldsymbol{\varepsilon}}{\sqrt{t}}, \\ \partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) &= -\frac{d}{2t} + \frac{\|\boldsymbol{x} - \boldsymbol{x}_0\|^2}{2t^2} = \frac{\|\boldsymbol{\varepsilon}\|^2 - d}{2t}. \end{aligned} \tag{17}$$

The corrupted marginal distribution is

$$p(\boldsymbol{x} \mid t) = \int q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\, p_{\mathrm{data}}(\boldsymbol{x}_0)\, \mathrm{d}\boldsymbol{x}_0, \tag{18}$$

where $p_{\mathrm{data}}(\boldsymbol{x}_0)$ is independent of t.

Given a noise-variance prior $p(t)$, the corresponding joint densities are

$$\begin{aligned} q(\boldsymbol{x}, t \mid \boldsymbol{x}_0) &= q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\, p(t), \\ p(\boldsymbol{x}, t) &= p(\boldsymbol{x} \mid t)\, p(t). \end{aligned} \tag{19}$$

We repeatedly use

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}[\boldsymbol{\varepsilon}] &= \mathbf{0}, & \mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\top] &= \boldsymbol{I}_d, \\ \mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}[\|\boldsymbol{\varepsilon}\|^2] &= d, & \mathrm{Var}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}(\|\boldsymbol{\varepsilon}\|^2) &= 2d, \\ \mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}[\boldsymbol{\varepsilon}\|\boldsymbol{\varepsilon}\|^2] &= \mathbf{0}, & \mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}[(\boldsymbol{\varepsilon}^\top \boldsymbol{a})^2] &= \|\boldsymbol{a}\|^2, \end{aligned} \tag{20}$$

for any fixed $\boldsymbol{a} \in \mathbb{R}^d$.

A.2 Proof of Score Scaling (Proposition 2.1)

We show that

$$\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|t)}\left[(\partial_{\boldsymbol{x}_i} \log p(\boldsymbol{x} \mid t))^2\right] \le \frac{1}{t}, \qquad \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|t)}\left[(\partial_t \log p(\boldsymbol{x} \mid t))^2\right] \le \frac{d}{2t^2}. \tag{21}$$

Proof. For the spatial score, we first use the posterior score identity

$$\partial_{x_i} \log p(\boldsymbol{x} \mid t) = \mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0|\boldsymbol{x},t)}\left[\partial_{x_i} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\right]. \tag{22}$$

Therefore,

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|t)}\left[(\partial_{x_i} \log p(\boldsymbol{x} \mid t))^2\right] \\ &\overset{\text{(i)}}{\le} \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|t)}\left[\mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0|\boldsymbol{x},t)}\left[(\partial_{x_i} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t))^2\right]\right] \\ &\overset{\text{(ii)}}{=} \mathbb{E}_{\substack{\boldsymbol{x}_0\sim p(\boldsymbol{x}_0) \\ \boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{x}_0,t)}}\left[(\partial_{x_i} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t))^2\right] \\ &\overset{\text{(iii)}}{=} \frac{1}{t}, \end{aligned} \tag{23}$$

where (i) follows from Jensen's inequality, (ii) from the law of total expectation, and (iii) from eqs. (17) and (20).

For the variance score, since $p(\boldsymbol{x}_0)$ does not depend on t, the analogous posterior score identity gives

$$\partial_t \log p(\boldsymbol{x} \mid t) = \mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0|\boldsymbol{x},t)}\left[\partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\right]. \tag{24}$$

Hence,

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|t)}\left[(\partial_t \log p(\boldsymbol{x} \mid t))^2\right] \\ &\overset{\text{(i)}}{\le} \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|t)}\left[\mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0|\boldsymbol{x},t)}\left[(\partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t))^2\right]\right] \\ &\overset{\text{(ii)}}{=} \mathbb{E}_{\substack{\boldsymbol{x}_0\sim p(\boldsymbol{x}_0) \\ \boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{x}_0,t)}}\left[(\partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t))^2\right] \\ &\overset{\text{(iii)}}{=} \frac{1}{4t^2}\mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}\left[\left(\|\boldsymbol{\varepsilon}\|^2 - d\right)^2\right] \\ &\overset{\text{(iv)}}{=} \frac{1}{4t^2}\mathrm{Var}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}(\|\boldsymbol{\varepsilon}\|^2) = \frac{d}{2t^2}, \end{aligned} \tag{25}$$

where (i) follows from Jensen's inequality, (ii) from the law of total expectation, (iii) from the definition of the variance, and eq. (17), and (iv) from eq. (20).

Thus,

$$\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|t)}\left[(\partial_{x_i} \log p(\boldsymbol{x} \mid t))^2\right] \le \frac{1}{t}, \qquad \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|t)}\left[(\partial_t \log p(\boldsymbol{x} \mid t))^2\right] \le \frac{d}{2t^2}, \tag{26}$$

which proves the claim. □

A.3 Proof of Fisher Metric (Proposition 2.2)

We show that the corruption-induced Fisher metric for $q(\boldsymbol{x}, t \mid \boldsymbol{x}_0)$ given by eq. (15) at fixed t is

$$\boldsymbol{\mathcal{I}}(t) = \begin{pmatrix} t^{-1}\boldsymbol{I}_d & \mathbf{0} \\ \mathbf{0}^\top & \dfrac{d}{2t^2} + (\partial_t \log p(t))^2 \end{pmatrix}. \tag{27}$$

Proof. We compute the individual blocks of the Fisher information matrix

$$\boldsymbol{\mathcal{I}}_{ij} = \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}\mid\boldsymbol{x}_0,t)}\left[\partial_i \log q(\boldsymbol{x}, t \mid \boldsymbol{x}_0)\, \partial_j \log q(\boldsymbol{x}, t \mid \boldsymbol{x}_0)\right], \tag{28}$$

where i, j index the coordinates of $(\boldsymbol{x}, t)$.

From eq. (19),

$$\begin{aligned} \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x}, t \mid \boldsymbol{x}_0) &= \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t), \\ \partial_t \log q(\boldsymbol{x}, t \mid \boldsymbol{x}_0) &= \partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) + \partial_t \log p(t), \end{aligned} \tag{29}$$

where the first equality follows from the independence of $p(t)$ from $\boldsymbol{x}$.

Spatial block. We obtain

$$\begin{aligned} \boldsymbol{\mathcal{I}}_{xx} &= \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}\mid\boldsymbol{x}_0,t)}\left[\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)^\top\right] \\ &\overset{\text{(i)}}{=} \mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}\left[\frac{\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\top}{t}\right] \\ &\overset{\text{(ii)}}{=} \frac{1}{t}\boldsymbol{I}_d. \end{aligned} \tag{30}$$

Here, (i) follows from eqs. (16) and (17), and (ii) from eq. (20).

Noise-variance block. First, the conditional variance score has zero mean:

$$\begin{aligned} \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}\mid\boldsymbol{x}_0,t)}\left[\partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\right] &\overset{\text{(i)}}{=} \mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}\left[\frac{\|\boldsymbol{\varepsilon}\|^2 - d}{2t}\right] \\ &\overset{\text{(ii)}}{=} 0. \end{aligned} \tag{31}$$

Here, (i) follows from eq. (17), and (ii) from eq. (20). Therefore,

$$\begin{aligned} \boldsymbol{\mathcal{I}}_{tt} &= \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}\mid\boldsymbol{x}_0,t)}\left[\left(\partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) + \partial_t \log p(t)\right)^2\right] \\ &\overset{\text{(i)}}{=} \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}\mid\boldsymbol{x}_0,t)}\left[\left(\partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\right)^2\right] + \left(\partial_t \log p(t)\right)^2 \\ &\overset{\text{(ii)}}{=} \frac{1}{4t^2}\mathrm{Var}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}\left(\|\boldsymbol{\varepsilon}\|^2\right) + \left(\partial_t \log p(t)\right)^2 \\ &\overset{\text{(iii)}}{=} \frac{d}{2t^2} + \left(\partial_t \log p(t)\right)^2. \end{aligned} \tag{32}$$

Here, (i) uses the zero-mean conditional noise-variance score above, (ii) follows from eq. (17), and (iii) from eq. (20).

Cross block. Finally,

$$\begin{aligned} \boldsymbol{\mathcal{I}}_{xt} &= \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}\mid\boldsymbol{x}_0,t)}\left[\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\left(\partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) + \partial_t \log p(t)\right)\right] \\ &\overset{\text{(i)}}{=} \mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}\left[-\frac{\boldsymbol{\varepsilon}}{\sqrt{t}}\left(\frac{\|\boldsymbol{\varepsilon}\|^2 - d}{2t} + \partial_t \log p(t)\right)\right] \\ &\overset{\text{(ii)}}{=} \mathbf{0}. \end{aligned} \tag{33}$$

Here, (i) follows from eq. (17), and (ii) from $\mathbb{E}[\boldsymbol{\varepsilon}] = \mathbf{0}$ and $\mathbb{E}[\boldsymbol{\varepsilon}\|\boldsymbol{\varepsilon}\|^2] = \mathbf{0}$ by symmetry.

Thus, $\boldsymbol{\mathcal{I}}_{xt} = \mathbf{0}$ and $\boldsymbol{\mathcal{I}}_{tx} = \boldsymbol{\mathcal{I}}_{xt}^\top = \mathbf{0}^\top$, which proves the result. □

A.4 Proof of Fisher Metric in Log Variance (Proposition 2.3)

We show that under the coordinate transformation $\tau = \log t$, the Fisher metric in eq. (9) becomes

$$\boldsymbol{\mathcal{I}}(\tau) = \begin{pmatrix} e^{-\tau}\boldsymbol{I}_d & \mathbf{0} \\ \mathbf{0}^\top & \frac{d}{2} + (\partial_\tau \log p_\tau(\tau))^2 \end{pmatrix}. \tag{34}$$

Equivalently, expressed in terms of the original noise-variance coordinate t,

$$\boldsymbol{\mathcal{I}}(\tau)\big|_{\tau=\log t} = \begin{pmatrix} t^{-1}\boldsymbol{I}_d & \mathbf{0} \\ \mathbf{0}^\top & \frac{d}{2} + (1 + t\,\partial_t \log p(t))^2 \end{pmatrix}. \tag{35}$$

Proof. Let $\tau = \log t$, such that $t = e^\tau$. Under this change of variables, the variance prior transforms as

$$p_\tau(\tau) = p(t)\left|\frac{\mathrm{d}t}{\mathrm{d}\tau}\right| = e^\tau p(e^\tau), \tag{36}$$

and therefore

$$q(\boldsymbol{x}, \tau \mid \boldsymbol{x}_0) = q(\boldsymbol{x} \mid \boldsymbol{x}_0, e^\tau)\, p_\tau(\tau). \tag{37}$$

Since $\partial_\tau = t\,\partial_t$, the corresponding score vector is

$$\nabla_{(\boldsymbol{x},\tau)} \log q(\boldsymbol{x}, \tau \mid \boldsymbol{x}_0) = \begin{pmatrix} \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) \\ t\,\partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) + \partial_\tau \log p_\tau(\tau) \end{pmatrix}. \tag{38}$$

Using the definition of the Fisher metric in eq. (28), the spatial block is unchanged apart from the substitution $t = e^\tau$:

$$\boldsymbol{\mathcal{I}}_{xx}(\tau) = \boldsymbol{\mathcal{I}}_{xx}(t)\big|_{t=e^\tau} = e^{-\tau}\boldsymbol{I}_d. \tag{39}$$

For the variance block,

$$\begin{aligned} \boldsymbol{\mathcal{I}}_{\tau\tau} &= \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{x}_0,t)}\left[(t\,\partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) + \partial_\tau \log p_\tau(\tau))^2\right] \\ &\overset{\text{(i)}}{=} t^2 \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{x}_0,t)}\left[(\partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t))^2\right] + (\partial_\tau \log p_\tau(\tau))^2 \\ &\overset{\text{(ii)}}{=} \frac{d}{2} + (\partial_\tau \log p_\tau(\tau))^2. \end{aligned} \tag{40}$$

Here, (i) uses the zero-mean conditional noise-variance score eq. (31), which eliminates the cross term, and (ii) follows from the conditional contribution of the $\boldsymbol{\mathcal{I}}_{tt}$ eq. (32).

For the cross block,

$$\begin{aligned} \boldsymbol{\mathcal{I}}_{x\tau} &= \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{x}_0,t)}\left[\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\,(t\,\partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) + \partial_\tau \log p_\tau(\tau))\right] \\ &\overset{\text{(i)}}{=} t\,\boldsymbol{\mathcal{I}}_{xt} + \partial_\tau \log p_\tau(\tau)\, \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{x}_0,t)}\left[\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\right] \\ &\overset{\text{(ii)}}{=} \mathbf{0}. \end{aligned} \tag{41}$$

Here, (i) follows by linearity of expectation, and (ii) uses $\boldsymbol{\mathcal{I}}_{xt} = \mathbf{0}$ from eq. (9) together with the zero-mean score identity eq. (31). By symmetry, $\boldsymbol{\mathcal{I}}_{\tau x} = \mathbf{0}^\top$.

Hence,

$$\boldsymbol{\mathcal{I}}(\tau) = \begin{pmatrix} e^{-\tau}\boldsymbol{I}_d & \mathbf{0} \\ \mathbf{0}^\top & \dfrac{d}{2} + (\partial_\tau \log p_\tau(\tau))^2 \end{pmatrix}. \tag{42}$$

Finally,

$$\begin{aligned} \partial_\tau \log p_\tau(\tau) &\overset{\text{(i)}}{=} t\,\partial_t \log(t\,p(t)) \\ &\overset{\text{(ii)}}{=} 1 + t\,\partial_t \log p(t), \qquad t = e^\tau, \end{aligned} \tag{43}$$

where (i) follows from $p_\tau(\tau) = t\,p(t)$ and $\partial_\tau = t\,\partial_t$, and (ii) from the logarithm rules. Substituting this expression yields the equivalent form in terms of the original noise-variance coordinate. □

A.5 Curvature Preconditioner

We show that the Hessian of the marginal distribution in eq. (18), with respect to the joint variable $\boldsymbol{z} = (\boldsymbol{x}, t)$, is given by

$$\begin{aligned}\nabla_{\boldsymbol{z}}^2 \log p(\boldsymbol{x} \mid t) =& \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0 \mid \boldsymbol{x}, t)} \left[\nabla_{\boldsymbol{z}}^2 \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\right] \\ &+ \mathrm{Cov}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0 \mid \boldsymbol{x}, t)} \left(\nabla_{\boldsymbol{z}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\right).\end{aligned} \tag{44}$$

We further evaluate the expression by assuming the VE corruption process in eq. (15) and an underlying GMM data distribution as defined in eq. (115).

The Hessian of the full joint density $p(\boldsymbol{x}, t) = p(\boldsymbol{x} \mid t)p(t)$ is obtained by adding the prior curvature $\partial_t^2 \log p(t)$ to the tt-block of $\nabla_{\boldsymbol{z}}^2 \log p(\boldsymbol{x} \mid t)$, while the remaining blocks are unchanged. This follows because the prior $p(t)$ is independent of $\boldsymbol{x}$, so all Hessian blocks involving derivatives with respect to $\boldsymbol{x}$ vanish.

Proof. Starting from eq. (18), we first obtain the score identity

$$\begin{aligned}\nabla_{\boldsymbol{z}} \log p(\boldsymbol{x} \mid t) &= \frac{\nabla_{\boldsymbol{z}} p(\boldsymbol{x} \mid t)}{p(\boldsymbol{x} \mid t)} \\ &\overset{\text{(i)}}{=} \frac{\int p(\boldsymbol{x}_0)\, \nabla_{\boldsymbol{z}} q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\, \mathrm{d}\boldsymbol{x}_0}{p(\boldsymbol{x} \mid t)} \\ &\overset{\text{(ii)}}{=} \frac{\int p(\boldsymbol{x}_0)\, q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\, \nabla_{\boldsymbol{z}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\, \mathrm{d}\boldsymbol{x}_0}{p(\boldsymbol{x} \mid t)} \\ &\overset{\text{(iii)}}{=} \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0 \mid \boldsymbol{x}, t)} \left[\nabla_{\boldsymbol{z}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\right],\end{aligned} \tag{45}$$

where (i) follows by exchanging differentiation and integration, (ii) uses $\nabla_{\boldsymbol{z}} q = q \nabla_{\boldsymbol{z}} \log q$, and (iii) follows from Bayes' rule.

Differentiating once more with respect to $\boldsymbol{z}$ gives

$$\begin{aligned}\nabla_{\boldsymbol{z}}^2 \log p(\boldsymbol{x} \mid t) &= \nabla_{\boldsymbol{z}} \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0 \mid \boldsymbol{x}, t)} \left[\nabla_{\boldsymbol{z}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\right] \\ &\overset{\text{(i)}}{=} \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0 \mid \boldsymbol{x}, t)} \left[\nabla_{\boldsymbol{z}}^2 \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\right] \\ &\quad + \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0 \mid \boldsymbol{x}, t)} \left[\nabla_{\boldsymbol{z}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) \left(\nabla_{\boldsymbol{z}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) - \nabla_{\boldsymbol{z}} \log p(\boldsymbol{x} \mid t)\right)^\top\right] \\ &\overset{\text{(ii)}}{=} \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0 \mid \boldsymbol{x}, t)} \left[\nabla_{\boldsymbol{z}}^2 \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\right] \\ &\quad + \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0 \mid \boldsymbol{x}, t)} \left[\nabla_{\boldsymbol{z}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) \nabla_{\boldsymbol{z}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)^\top\right] \\ &\quad - \nabla_{\boldsymbol{z}} \log p(\boldsymbol{x} \mid t) \nabla_{\boldsymbol{z}} \log p(\boldsymbol{x} \mid t)^\top \\ &\overset{\text{(iii)}}{=} \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0 \mid \boldsymbol{x}, t)} \left[\nabla_{\boldsymbol{z}}^2 \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\right] + \mathrm{Cov}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0 \mid \boldsymbol{x}, t)} \left(\nabla_{\boldsymbol{z}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\right).\end{aligned} \tag{46}$$

Here, (i) follows from the product rule and the identity $\nabla_{\boldsymbol{z}} \log q(\boldsymbol{x}_0 \mid \boldsymbol{x}, t) = \nabla_{\boldsymbol{z}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) - \nabla_{\boldsymbol{z}} \log p(\boldsymbol{x} \mid t)$, which follows from Bayes' rule and the independence of $p(\boldsymbol{x}_0)$ from $\boldsymbol{z}$. Furthermore, (ii) uses the posterior score identity derived above, and (iii) follows from the definition of the covariance. □

Restricting to the VE corruption process eq. (15). Using the conditional scores in eq. (17), the required second derivatives of the VE corruption kernel are

$$\begin{aligned}\nabla_{\boldsymbol{x}}^2 \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) &= -\frac{1}{t} \boldsymbol{I}_d, \\ \partial_t \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) &= \frac{\boldsymbol{x} - \boldsymbol{x}_0}{t^2}, \\ \partial_t^2 \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) &= \frac{d}{2t^2} - \frac{\|\boldsymbol{x} - \boldsymbol{x}_0\|^2}{t^3}.\end{aligned} \tag{47}$$

Substituting these expressions together with eq. (17) into eq. (44) gives

$$\begin{aligned}
\nabla_{\boldsymbol{x}}^2 \log p(\boldsymbol{x} \mid t) &= -\frac{1}{t}\boldsymbol{I}_d + \frac{1}{t^2}\mathrm{Cov}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0\mid\boldsymbol{x},t)}\left(\boldsymbol{x}_0\right), \\
\partial_t \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x} \mid t) &= \frac{1}{t^2}\mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0\mid\boldsymbol{x},t)}\left[\boldsymbol{x}-\boldsymbol{x}_0\right] - \frac{1}{2t^3}\mathrm{Cov}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0\mid\boldsymbol{x},t)}\left(\boldsymbol{x}-\boldsymbol{x}_0,\, \|\boldsymbol{x}-\boldsymbol{x}_0\|^2\right), \\
\left(\nabla_{\boldsymbol{x}}\partial_t \log p(\boldsymbol{x} \mid t)\right)^\top &= \left(\partial_t \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x} \mid t)\right)^\top, \\
\partial_t^2 \log p(\boldsymbol{x} \mid t) &= \frac{d}{2t^2} - \frac{1}{t^3}\mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0\mid\boldsymbol{x},t)}\left[\|\boldsymbol{x}-\boldsymbol{x}_0\|^2\right] + \frac{1}{4t^4}\mathrm{Var}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0\mid\boldsymbol{x},t)}\left(\|\boldsymbol{x}-\boldsymbol{x}_0\|^2\right).
\end{aligned} \tag{48}$$

Restricting to a GMM data distribution. For the GMM data distribution in eq. (115), the posterior $q(\boldsymbol{x}_0 \mid \boldsymbol{x}, t)$ is again a Gaussian mixture,

$$q(\boldsymbol{x}_0 \mid \boldsymbol{x}, t) = \sum_{k=1}^{K} \gamma_k(\boldsymbol{x}, t)\, \mathcal{N}\left(\boldsymbol{x}_0 \mid \boldsymbol{m}_k, c_k \boldsymbol{I}_d\right), \tag{49}$$

where

$$\begin{aligned}
\gamma_k(\boldsymbol{x}, t) &= \frac{w_k\, \mathcal{N}\big(\boldsymbol{x} \mid \boldsymbol{\mu}_k, (\sigma_k^2 + t)\boldsymbol{I}_d\big)}{\sum_{j=1}^{K} w_j\, \mathcal{N}\big(\boldsymbol{x} \mid \boldsymbol{\mu}_j, (\sigma_j^2 + t)\boldsymbol{I}_d\big)}, \\
\boldsymbol{m}_k &= \frac{t\boldsymbol{\mu}_k + \sigma_k^2 \boldsymbol{x}}{\sigma_k^2 + t}, \qquad c_k = \frac{\sigma_k^2 t}{\sigma_k^2 + t}.
\end{aligned} \tag{50}$$

Defining

$$\begin{aligned}
&\boldsymbol{\delta}_k = \boldsymbol{x} - \boldsymbol{m}_k, \qquad \bar{\boldsymbol{m}} = \sum_{k=1}^{K} \gamma_k \boldsymbol{m}_k, \quad \bar{\boldsymbol{\delta}} = \sum_{k=1}^{K} \gamma_k \boldsymbol{\delta}_k, \\
&Q_k = \|\boldsymbol{\delta}_k\|^2 + d\, c_k, \quad \bar{Q} = \sum_{k=1}^{K} \gamma_k Q_k,
\end{aligned} \tag{51}$$

the posterior moments appearing in eq. (48) are

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0\mid\boldsymbol{x},t)}\left[\boldsymbol{x}-\boldsymbol{x}_0\right] &= \bar{\boldsymbol{\delta}}, \\
\mathrm{Cov}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0\mid\boldsymbol{x},t)}\left(\boldsymbol{x}_0\right) &= \sum_{k=1}^{K} \gamma_k \left[c_k \boldsymbol{I}_d + (\boldsymbol{m}_k - \bar{\boldsymbol{m}})(\boldsymbol{m}_k - \bar{\boldsymbol{m}})^\top\right], \\
\mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0\mid\boldsymbol{x},t)}\left[\|\boldsymbol{x}-\boldsymbol{x}_0\|^2\right] &= \bar{Q}, \\
\mathrm{Cov}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0\mid\boldsymbol{x},t)}\left(\boldsymbol{x}-\boldsymbol{x}_0,\, \|\boldsymbol{x}-\boldsymbol{x}_0\|^2\right) &= \sum_{k=1}^{K} \gamma_k \left[2c_k \boldsymbol{\delta}_k + (\boldsymbol{\delta}_k - \bar{\boldsymbol{\delta}})(Q_k - \bar{Q})\right], \\
\mathrm{Var}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0\mid\boldsymbol{x},t)}\left(\|\boldsymbol{x}-\boldsymbol{x}_0\|^2\right) &= \sum_{k=1}^{K} \gamma_k \left[2d\, c_k^2 + 4c_k \|\boldsymbol{\delta}_k\|^2 + (Q_k - \bar{Q})^2\right].
\end{aligned} \tag{52}$$

Thus, for a GMM data distribution, all posterior moments in eq. (48) reduce to finite sums over the mixture components, yielding an exact closed-form expression for the marginal Hessian.

B Extension of the Geometry Characterization to the VP Kernel

B.1 Preliminaries

Throughout this section, we consider the Gaussian VP corruption kernel parameterized directly by the noise variance $t \in (0, 1)$,

$$q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) \coloneqq \mathcal{N}\left(\boldsymbol{x} \mid \sqrt{1-t}\,\boldsymbol{x}_0, t\,\boldsymbol{I}_d\right) = \frac{1}{(2\pi t)^{d/2}} \exp\left(-\frac{\|\boldsymbol{x} - \sqrt{1-t}\,\boldsymbol{x}_0\|^2}{2t}\right), \tag{53}$$

which can equivalently be defined via

$$\boldsymbol{x} = \sqrt{1-t}\,\boldsymbol{x}_0 + \sqrt{t}\,\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d). \tag{54}$$

The conditional spatial and variance scores of the corruption kernel are

$$\begin{aligned} \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) &= -\frac{\boldsymbol{x} - \sqrt{1-t}\,\boldsymbol{x}_0}{t} = -\frac{\boldsymbol{\varepsilon}}{\sqrt{t}}, \\ \partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) &= -\frac{d}{2t} + \frac{\|\boldsymbol{x} - \sqrt{1-t}\,\boldsymbol{x}_0\|^2}{2t^2} - \frac{\left(\boldsymbol{x} - \sqrt{1-t}\,\boldsymbol{x}_0\right)^\top \boldsymbol{x}_0}{2t\sqrt{1-t}} \\ &= \frac{\|\boldsymbol{\varepsilon}\|^2 - d}{2t} - \frac{\boldsymbol{\varepsilon}^\top \boldsymbol{x}_0}{2\sqrt{t(1-t)}}. \end{aligned} \tag{55}$$

The corrupted marginal distribution and the joint densities are given in eq. (18) and eq. (19) respectively.

B.2 Score Scaling for the VP Corruption Kernel

We show that

$$\begin{aligned} \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|t)} \left[\left(\partial_{x_i} \log p(\boldsymbol{x} \mid t)\right)^2 \right] &\le \frac{1}{t}, \\ \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|t)} \left[\left(\partial_t \log p(\boldsymbol{x} \mid t)\right)^2 \right] &\le \frac{d}{2t^2} + \frac{\mathbb{E}_{\boldsymbol{x}_0\sim p_{\mathrm{data}}}[\|\boldsymbol{x}_0\|^2]}{4t(1-t)}. \end{aligned} \tag{56}$$

Note that under the assumption that the data are normalized to zero mean and unit variance in each dimension $\mathbb{E}_{\boldsymbol{x}_0\sim p_{\mathrm{data}}}[\|\boldsymbol{x}_0\|^2] = d$.

Proof. For the spatial score, we first use the posterior score identity

$$\partial_{x_i} \log p(\boldsymbol{x} \mid t) = \mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0|\boldsymbol{x},t)} \left[\partial_{x_i} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) \right]. \tag{57}$$

Therefore,

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|t)} \left[\left(\partial_{x_i} \log p(\boldsymbol{x} \mid t)\right)^2 \right] \\ &\overset{\text{(i)}}{\le} \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|t)} \left[\mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0|\boldsymbol{x},t)} \left[\left(\partial_{x_i} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\right)^2 \right] \right] \\ &\overset{\text{(ii)}}{=} \mathbb{E}_{\substack{\boldsymbol{x}_0\sim p_{\mathrm{data}} \\ \boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{x}_0,t)}} \left[\left(\partial_{x_i} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\right)^2 \right] \\ &\overset{\text{(iii)}}{=} \frac{1}{t}, \end{aligned} \tag{58}$$

where (i) follows from Jensen's inequality, (ii) from the law of total expectation, and (iii) from eqs. (20) and (55).

For the noise-variance score, since $p_{\mathrm{data}}(\boldsymbol{x}_0)$ does not depend on t, the analogous posterior score identity gives

$$\partial_t \log p(\boldsymbol{x} \mid t) = \mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0|\boldsymbol{x},t)} \left[\partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) \right]. \tag{59}$$

Hence,

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|t)}\left[(\partial_t \log p(\boldsymbol{x}\mid t))^2\right] \\
&\overset{\text{(i)}}{\leq} \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|t)}\left[\mathbb{E}_{\boldsymbol{x}_0\sim q(\boldsymbol{x}_0|\boldsymbol{x},t)}\left[(\partial_t \log q(\boldsymbol{x}\mid \boldsymbol{x}_0,t))^2\right]\right] \\
&\overset{\text{(ii)}}{=} \mathbb{E}_{\substack{\boldsymbol{x}_0\sim p_{\text{data}}\\ \boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{x}_0,t)}}\left[(\partial_t \log q(\boldsymbol{x}\mid \boldsymbol{x}_0,t))^2\right] \\
&\overset{\text{(iii)}}{=} \mathbb{E}_{\substack{\boldsymbol{x}_0\sim p_{\text{data}}\\ \boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}}\left[\left(\frac{\|\boldsymbol{\varepsilon}\|^2-d}{2t}-\frac{\boldsymbol{\varepsilon}^\top\boldsymbol{x}_0}{2\sqrt{t(1-t)}}\right)^2\right] \\
&\overset{\text{(iv)}}{=} \frac{1}{4t^2}\mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}\left[\left(\|\boldsymbol{\varepsilon}\|^2-d\right)^2\right] \\
&\qquad + \frac{1}{4t(1-t)}\mathbb{E}_{\substack{\boldsymbol{x}_0\sim p_{\text{data}}\\ \boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}}\left[\left(\boldsymbol{\varepsilon}^\top\boldsymbol{x}_0\right)^2\right] \\
&\overset{\text{(v)}}{=} \frac{1}{4t^2}\mathrm{Var}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}(\|\boldsymbol{\varepsilon}\|^2)+\frac{\mathbb{E}_{\boldsymbol{x}_0\sim p_{\text{data}}}[\|\boldsymbol{x}_0\|^2]}{4t(1-t)} \\
&= \frac{d}{2t^2}+\frac{\mathbb{E}_{\boldsymbol{x}_0\sim p_{\text{data}}}[\|\boldsymbol{x}_0\|^2]}{4t(1-t)}.
\end{aligned} \tag{60}$$

Here, (i) follows from Jensen's inequality, (ii) from the law of total expectation, and (iii) from eqs. (54) and (55). In (iv), the mixed term vanishes by eq. (20). Finally, (v) follows from eq. (20). □

B.3 Corruption-Induced Fisher Metric for the VP Kernel

We show that the corruption-induced Fisher metric for $q(\boldsymbol{x}, t \mid \boldsymbol{x}_0)$ given by eq. (53) at fixed $\boldsymbol{x}_0$ and t is

$$\boldsymbol{\mathcal{I}}(\boldsymbol{x}_0,t)=\begin{pmatrix} t^{-1}\boldsymbol{I}_d & \dfrac{\boldsymbol{x}_0}{2t\sqrt{1-t}} \\ \dfrac{\boldsymbol{x}_0^\top}{2t\sqrt{1-t}} & \dfrac{d}{2t^2}+\dfrac{\|\boldsymbol{x}_0\|^2}{4t(1-t)}+(\partial_t \log p(t))^2 \end{pmatrix}. \tag{61}$$

Proof. We compute the individual blocks of the Fisher information matrix

$$\boldsymbol{\mathcal{I}}_{ij}=\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{x}_0,t)}\left[\partial_i \log q(\boldsymbol{x},t\mid\boldsymbol{x}_0)\,\partial_j \log q(\boldsymbol{x},t\mid\boldsymbol{x}_0)\right], \tag{62}$$

where i, j index the coordinates of $(\boldsymbol{x}, t)$.

From eq. (19),

$$\begin{aligned}
\nabla_{\boldsymbol{x}}\log q(\boldsymbol{x},t\mid\boldsymbol{x}_0)&=\nabla_{\boldsymbol{x}}\log q(\boldsymbol{x}\mid\boldsymbol{x}_0,t),\\
\partial_t \log q(\boldsymbol{x},t\mid\boldsymbol{x}_0)&=\partial_t \log q(\boldsymbol{x}\mid\boldsymbol{x}_0,t)+\partial_t \log p(t),
\end{aligned} \tag{63}$$

where the first equality follows from the independence of $p(t)$ from $\boldsymbol{x}$.

Spatial block. We obtain

$$\begin{aligned}
\boldsymbol{\mathcal{I}}_{xx}&=\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{x}_0,t)}\left[\nabla_{\boldsymbol{x}}\log q(\boldsymbol{x}\mid\boldsymbol{x}_0,t)\nabla_{\boldsymbol{x}}\log q(\boldsymbol{x}\mid\boldsymbol{x}_0,t)^\top\right]\\
&\overset{\text{(i)}}{=}\mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}\left[\frac{\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\top}{t}\right]\\
&\overset{\text{(ii)}}{=}\frac{1}{t}\boldsymbol{I}_d.
\end{aligned} \tag{64}$$

Here, (i) follows from eqs. (54) and (55), and (ii) from eq. (20).

Noise-variance block. First, the conditional variance score has zero mean:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{x}_0,t)}\left[\partial_t \log q(\boldsymbol{x}\mid \boldsymbol{x}_0,t)\right] \\
&\overset{\text{(i)}}{=} \mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}\left[\frac{\|\boldsymbol{\varepsilon}\|^2-d}{2t}-\frac{\boldsymbol{\varepsilon}^\top \boldsymbol{x}_0}{2\sqrt{t(1-t)}}\right] \\
&\overset{\text{(ii)}}{=} 0.
\end{aligned} \tag{65}$$

Here, (i) follows from eq. (55), and (ii) from eq. (20).

Therefore,

$$\begin{aligned}
\boldsymbol{\mathcal{I}}_{tt} &= \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{x}_0,t)}\left[\left(\partial_t \log q(\boldsymbol{x}\mid \boldsymbol{x}_0,t)+\partial_t \log p(t)\right)^2\right] \\
&\overset{\text{(i)}}{=} \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{x}_0,t)}\left[\left(\partial_t \log q(\boldsymbol{x}\mid \boldsymbol{x}_0,t)\right)^2\right]+\left(\partial_t \log p(t)\right)^2 \\
&\overset{\text{(ii)}}{=} \frac{1}{4t^2}\mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}\left[\left(\|\boldsymbol{\varepsilon}\|^2-d\right)^2\right] \\
&\qquad+\frac{1}{4t(1-t)}\mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}\left[\left(\boldsymbol{\varepsilon}^\top\boldsymbol{x}_0\right)^2\right]+\left(\partial_t \log p(t)\right)^2 \\
&\overset{\text{(iii)}}{=} \frac{d}{2t^2}+\frac{\|\boldsymbol{x}_0\|^2}{4t(1-t)}+\left(\partial_t \log p(t)\right)^2.
\end{aligned} \tag{66}$$

Here, (i) uses the zero-mean conditional temporal score above. In (ii), the mixed term vanishes by eq. (20), and (iii) follows from eq. (20).

Cross block. Finally,

$$\begin{aligned}
\boldsymbol{\mathcal{I}}_{xt} &= \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x}|\boldsymbol{x}_0,t)}\left[\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x}\mid \boldsymbol{x}_0,t)\left(\partial_t \log q(\boldsymbol{x}\mid \boldsymbol{x}_0,t)+\partial_t \log p(t)\right)\right] \\
&\overset{\text{(i)}}{=} \mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}\left[-\frac{\boldsymbol{\varepsilon}}{\sqrt{t}}\left(\frac{\|\boldsymbol{\varepsilon}\|^2-d}{2t}-\frac{\boldsymbol{\varepsilon}^\top\boldsymbol{x}_0}{2\sqrt{t(1-t)}}+\partial_t \log p(t)\right)\right] \\
&\overset{\text{(ii)}}{=} \frac{1}{2t\sqrt{1-t}}\mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}\left[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\top\right]\boldsymbol{x}_0 \\
&\overset{\text{(iii)}}{=} \frac{\boldsymbol{x}_0}{2t\sqrt{1-t}}.
\end{aligned} \tag{67}$$

Here, (i) follows from eq. (55). In (ii), the terms proportional to $\boldsymbol{\varepsilon}(\|\boldsymbol{\varepsilon}\|^2-d)$ and $\boldsymbol{\varepsilon}$ vanish by symmetry and eq. (20), and (iii) follows from $\mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^\top]=\boldsymbol{I}_d$.

Thus,

$$\boldsymbol{\mathcal{I}}_{xt}=\frac{\boldsymbol{x}_0}{2t\sqrt{1-t}},\qquad \boldsymbol{\mathcal{I}}_{tx}=\boldsymbol{\mathcal{I}}_{xt}^\top=\frac{\boldsymbol{x}_0^\top}{2t\sqrt{1-t}}, \tag{68}$$

and consequently

$$\boldsymbol{\mathcal{I}}(\boldsymbol{x}_0,t)=\begin{pmatrix} t^{-1}\boldsymbol{I}_d & \dfrac{\boldsymbol{x}_0}{2t\sqrt{1-t}} \\ \dfrac{\boldsymbol{x}_0^\top}{2t\sqrt{1-t}} & \dfrac{d}{2t^2}+\dfrac{\|\boldsymbol{x}_0\|^2}{4t(1-t)}+\left(\partial_t \log p(t)\right)^2 \end{pmatrix}, \tag{69}$$

which proves the result. □

B.3.1 Inverse of the VP Fisher Metric

The inverse of the VP Fisher metric can be obtained using the block-matrix inversion formula. For eq. (61) the Schur complement of the spatial block is $S(t)=d/(2t^2)+\left(\partial_t \log p(t)\right)^2$, since the term $\|\boldsymbol{x}_0\|^2/[4t(1-t)]$ cancels with the contribution from the cross blocks. Consequently,

$$\boldsymbol{\mathcal{I}}(\boldsymbol{x}_0,t)^{-1}=\begin{pmatrix} t\boldsymbol{I}_d+\dfrac{\boldsymbol{x}_0\boldsymbol{x}_0^\top}{4(1-t)S(t)} & -\dfrac{\boldsymbol{x}_0}{2\sqrt{1-t}\,S(t)} \\ -\dfrac{\boldsymbol{x}_0^\top}{2\sqrt{1-t}\,S(t)} & \dfrac{1}{S(t)} \end{pmatrix}. \tag{70}$$

Thus, despite the sample-dependent and non-diagonal structure of the VP Fisher metric, its Schur complement is independent of $\boldsymbol{x}_0$.

B.3.2 Assuming Normalized Data

Under the additional assumption that the data are normalized to zero mean and unit variance,

$$\mathbb{E}_{\boldsymbol{x}_0\sim p_{\text{data}}}[\boldsymbol{x}_0] = \mathbf{0}, \qquad \mathbb{E}_{\boldsymbol{x}_0\sim p_{\text{data}}}\left[\|\boldsymbol{x}_0\|^2\right] = d, \tag{71}$$

averaging the corruption-induced Fisher metric over the data distribution yields

$$\bar{\boldsymbol{\mathcal{I}}}(t) \coloneqq \mathbb{E}_{\boldsymbol{x}_0\sim p_{\text{data}}}\left[\boldsymbol{\mathcal{I}}(\boldsymbol{x}_0, t)\right] = \begin{pmatrix} t^{-1}\boldsymbol{I}_d & \mathbf{0} \\ \mathbf{0}^\top & \dfrac{d}{2t^2} + \dfrac{d}{4t(1-t)} + (\partial_t \log p(t))^2 \end{pmatrix}. \tag{72}$$

Consequently,

$$\bar{\boldsymbol{\mathcal{I}}}(t)^{-1} = \begin{pmatrix} t\boldsymbol{I}_d & \mathbf{0} \\ \mathbf{0}^\top & \left[\dfrac{d}{2t^2} + \dfrac{d}{4t(1-t)} + (\partial_t \log p(t))^2\right]^{-1} \end{pmatrix}. \tag{73}$$

Thus, under normalized and centered data, the data-averaged VP Fisher metric becomes diagonal and depends only on the noise coordinate t.

C Sampler Construction and Derivations

We consider the augmented state

$$\boldsymbol{z} = (\boldsymbol{x}, t) \in \mathcal{Z} := \mathbb{R}^d \times [t_{\min}, t_{\max}], \qquad \partial\mathcal{Z} = \mathbb{R}^d \times \{t_{\min}, t_{\max}\}. \tag{74}$$

Thus, only the noise-variance coordinate is bounded.

C.1 Derivation of DSDS SDE (Equation (12))

C.1.1 Divergence Correction for a State-Dependent Preconditioner

We first consider the dynamics in the interior of $\mathcal{Z}$ and summarize the derivation underlying Xifara et al. (2014, Eq. (9)); see their work for a more detailed treatment.

For a positive semidefinite, possibly state-dependent preconditioner $\boldsymbol{A}(\boldsymbol{z}) \in \mathbb{R}^{(d+1)\times(d+1)}$, consider

$$\mathrm{d}\boldsymbol{z}_s = \boldsymbol{b}(\boldsymbol{z}_s)\,\mathrm{d}s + \sqrt{2\boldsymbol{A}(\boldsymbol{z}_s)}\,\mathrm{d}\boldsymbol{w}_s. \tag{75}$$

Let $p_s(\boldsymbol{z})$ denote the density of $\boldsymbol{z}_s$. Its Fokker-Planck equation is

$$\partial_s p_s(\boldsymbol{z}) = -\nabla_{\boldsymbol{z}} \cdot \boldsymbol{j}_s(\boldsymbol{z}), \tag{76}$$

with probability current

$$\boldsymbol{j}_s(\boldsymbol{z}) = \boldsymbol{b}(\boldsymbol{z})p_s(\boldsymbol{z}) - \nabla_{\boldsymbol{z}} \cdot \left(\boldsymbol{A}(\boldsymbol{z})p_s(\boldsymbol{z})\right), \tag{77}$$

where the matrix divergence is defined as $\left(\nabla_{\boldsymbol{z}} \cdot \boldsymbol{A}(\boldsymbol{z})\right)_i = \sum_{j=1}^{d+1} \partial_{\boldsymbol{z}_j} A_{ij}(\boldsymbol{z})$.

Imposing zero probability current at stationarity, $\boldsymbol{j}(\boldsymbol{z}) = \mathbf{0}$ for $p_s = p$, gives

$$\begin{aligned} \boldsymbol{b}(\boldsymbol{z}) &= \frac{1}{p(\boldsymbol{z})}\nabla_{\boldsymbol{z}} \cdot \left(\boldsymbol{A}(\boldsymbol{z})p(\boldsymbol{z})\right) \\ &= \boldsymbol{A}(\boldsymbol{z})\nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}) + \nabla_{\boldsymbol{z}} \cdot \boldsymbol{A}(\boldsymbol{z}). \end{aligned} \tag{78}$$

Hence, the corrected Langevin dynamics are

$$\mathrm{d}\boldsymbol{z}_s = \left[\boldsymbol{A}(\boldsymbol{z}_s)\nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}_s) + \nabla_{\boldsymbol{z}} \cdot \boldsymbol{A}(\boldsymbol{z}_s)\right]\mathrm{d}s + \sqrt{2\boldsymbol{A}(\boldsymbol{z}_s)}\mathrm{d}\boldsymbol{w}_s, \tag{79}$$

which admit $p(\boldsymbol{z})$ as a stationary density.

C.1.2 Boundary Reflection

We next account for the boundary of $\mathcal{Z}$, following Lou & Ermon (2023, Appendix A.1, Eqs. (40)–(42)). They formulate the reflected dynamics in terms of the diffusion coefficient $\boldsymbol{\sigma}$; in our setting, $\boldsymbol{\sigma}(\boldsymbol{z}) = \sqrt{\boldsymbol{A}(\boldsymbol{z})}$ up to a constant factor. The reflected SDE is

$$\mathrm{d}\boldsymbol{z}_s = \boldsymbol{b}(\boldsymbol{z}_s)\,\mathrm{d}s + \sqrt{2\boldsymbol{A}(\boldsymbol{z}_s)}\,\mathrm{d}\boldsymbol{w}_s + \boldsymbol{v}(\boldsymbol{z}_s)\,\mathrm{d}L_s, \tag{80}$$

where L_s is a nondecreasing boundary regulator that increases only for $\boldsymbol{z}_s \in \partial\mathcal{Z}$. The reflection direction is the conormal direction

$$\boldsymbol{v}(\boldsymbol{z}) \propto \boldsymbol{A}(\boldsymbol{z})\boldsymbol{n}(\boldsymbol{z}), \tag{81}$$

where $\boldsymbol{n}$ denotes the inward unit normal, and any normalization factor in $\boldsymbol{v}$ is absorbed into the boundary regulator $\mathrm{d}L_s$. The corresponding reflected diffusion satisfies the no-flux condition

$$\boldsymbol{n}(\boldsymbol{z})^\top \boldsymbol{j}_s(\boldsymbol{z}) = 0, \qquad \boldsymbol{z} \in \partial\mathcal{Z}. \tag{82}$$

Since $\boldsymbol{j}(\boldsymbol{z}) = \boldsymbol{0}$ at stationarity by the interior construction above, the target density remains invariant.

For DSDS, only the noise-variance coordinate is bounded, and thus the inward unit normal is $+\boldsymbol{e}_t$ at $t_{\min}$ and $-\boldsymbol{e}_t$ at $t_{\max}$. Since the Fisher preconditioner in eq. (83) is block diagonal, $\boldsymbol{A}(t)\boldsymbol{n} = A_{tt}(t)\boldsymbol{n}$. Thus, the conormal and ordinary normal directions are parallel, and reflection acts only on t, leaving $\boldsymbol{x}$ unchanged. The same argument applies in log-variance coordinates.

C.1.3 Explicit Correction Terms for the Fisher Preconditioner

Divergence correction in normal time. For the joint state $\boldsymbol{z} = (\boldsymbol{x}, t)$, the Fisher preconditioner is

$$\boldsymbol{A}(t) = \begin{pmatrix} t\boldsymbol{I}_d & \boldsymbol{0} \\ \boldsymbol{0}^\top & a_t(t) \end{pmatrix}, \qquad a_t(t) = \left(\frac{d}{2t^2} + \left(\partial_t \log p(t)\right)^2\right)^{-1}. \tag{83}$$

Since the spatial block depends only on t and the off-diagonal blocks vanish,

$$\nabla_{\boldsymbol{z}} \cdot \boldsymbol{A}(t) = \begin{pmatrix} \boldsymbol{0} \\ \partial_t a_t(t) \end{pmatrix}. \tag{84}$$

Differentiating $a_t(t)$ gives

$$\partial_t a_t(t) = \frac{\dfrac{d}{t^3} - 2\,\partial_t \log p(t)\,\partial_t^2 \log p(t)}{\left(\dfrac{d}{2t^2} + \left(\partial_t \log p(t)\right)^2\right)^2}. \tag{85}$$

Divergence correction in log variance. In log-variance coordinates $\tau = \log t$, the preconditioner is

$$\boldsymbol{A}(\tau) = \begin{pmatrix} e^\tau \boldsymbol{I}_d & \boldsymbol{0} \\ \boldsymbol{0}^\top & a_\tau(\tau) \end{pmatrix}, \qquad a_\tau(\tau) = \left(\frac{d}{2} + \left(\partial_\tau \log p_\tau(\tau)\right)^2\right)^{-1}. \tag{86}$$

Since the spatial block again depends only on the time coordinate and the off-diagonal blocks vanish,

$$\nabla_{\boldsymbol{z}} \cdot \boldsymbol{A}(\tau) = \begin{pmatrix} \boldsymbol{0} \\ \partial_\tau a_\tau(\tau) \end{pmatrix}. \tag{87}$$

Differentiating $a_\tau(\tau)$ gives

$$\partial_\tau a_\tau(\tau) = -\frac{2\,\partial_\tau \log p_\tau(\tau)\,\partial_\tau^2 \log p_\tau(\tau)}{\left(\dfrac{d}{2} + \left(\partial_\tau \log p_\tau(\tau)\right)^2\right)^2}. \tag{88}$$

Using eq. (43) and $\partial_\tau^2 \log p_\tau(\tau) = t\,\partial_t \log p(t) + t^2\,\partial_t^2 \log p(t)$, we equivalently obtain

$$\partial_\tau a_\tau(\tau)\big|_{\tau=\log t} = -\frac{2t\,\left(1 + t\,\partial_t \log p(t)\right)\left(\partial_t \log p(t) + t\,\partial_t^2 \log p(t)\right)}{\left(\dfrac{d}{2} + \left(1 + t\,\partial_t \log p(t)\right)^2\right)^2}. \tag{89}$$

Again, no model retraining is required, as the only additional quantity involved is the Hessian of the prior $\partial_t^2 \log p(t)$, which is available in closed form.

Algorithm 3: Dual Score Diffusion Sampler (DSDS)

Input: energy model $U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t)$, log-variance prior $p_{\boldsymbol{\phi},\tau}(\tau)$, initial log-variance $\tau_0 \in [\tau_{\min}, \tau_{\max}]$, number of Langevin steps L, step size η, Fisher preconditioner $\boldsymbol{A}(\tau) = \mathrm{diag}(e^{\tau}\boldsymbol{I}_d, a_\tau(\tau))$ from eq. (86).
Output: sample $\boldsymbol{x}_L$

Initialize $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, t_{\max}\boldsymbol{I}_d)$
Set the dimensionally scaled temporal step size $\eta_\tau \leftarrow \sqrt{d}\,\eta$
for $\ell = 0$ **to** $L-1$ **do**
  Compute $\nabla_{\boldsymbol{x}} U_{\boldsymbol{\theta}}(\boldsymbol{x}_\ell \mid e^{\tau_\ell})$ and $\partial_\tau U_{\boldsymbol{\theta}}(\boldsymbol{x}_\ell \mid e^{\tau_\ell})$ eq. (10)
  Sample $\boldsymbol{\xi}_\ell \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$ and $\xi_\ell \sim \mathcal{N}(0,1)$
  $\boldsymbol{x}_{\ell+1} \leftarrow \boldsymbol{x}_\ell - \eta\, e^{\tau_\ell}\, \nabla_{\boldsymbol{x}} U_{\boldsymbol{\theta}}(\boldsymbol{x}_\ell \mid e^{\tau_\ell}) + \sqrt{2\eta e^{\tau_\ell}}\, \boldsymbol{\xi}_\ell$
  $\tilde{\tau}_{\ell+1} \leftarrow \tau_\ell + \eta_\tau \big[a_\tau(\tau_\ell)\,(-\partial_\tau U_{\boldsymbol{\theta}}(\boldsymbol{x}_\ell \mid e^{\tau_\ell}) + \partial_\tau \log p_{\boldsymbol{\phi},\tau}(\tau_\ell)) + \overbrace{\partial_\tau a_\tau}^{eq.\ (88)}(\tau_\ell)\big]$
    $+ \sqrt{2\eta_\tau a_\tau(\tau_\ell)}\,\xi_\ell$
  $\tau_{\ell+1} \leftarrow \mathrm{Reflect}\,(\tilde{\tau}_{\ell+1}; \tau_{\min}, \tau_{\max})$ eq. (91)
return $\boldsymbol{x}_L$

C.1.4 Euler-Maruyama Discretization and DSDS Algorithm

Finally, we discretize the reflected SDE using the Euler-Maruyama scheme with step size η. For the joint state $\boldsymbol{z}_\ell = (\boldsymbol{x}_\ell, t_\ell)$, the unconstrained update is

$$\tilde{\boldsymbol{z}}_{\ell+1} = \boldsymbol{z}_\ell + \eta\,[\boldsymbol{A}(\boldsymbol{z}_\ell)\nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}_\ell) + \nabla_{\boldsymbol{z}} \cdot \boldsymbol{A}(\boldsymbol{z}_\ell)] + \sqrt{2\eta \boldsymbol{A}(\boldsymbol{z}_\ell)}\boldsymbol{\xi}_\ell, \quad \boldsymbol{\xi}_\ell \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_{d+1}). \tag{90}$$

Since only the noise-variance coordinate is bounded, the spatial component is accepted directly, $\boldsymbol{x}_{\ell+1} = \tilde{\boldsymbol{x}}_{\ell+1}$, while the noise-variance coordinate is reflected back into its domain using the mirror reflection

$$\mathrm{Reflect}(\tilde{t}) = t_{\min} + (t_{\max} - t_{\min}) - \big|\big((\tilde{t} - t_{\min}) \bmod 2(t_{\max} - t_{\min})\big) - (t_{\max} - t_{\min})\big|\,. \tag{91}$$

The same discretization applies in log-variance coordinates after replacing t by τ.

Remark on the dimensionality scaling. Since the conditional noise-variance score scales as $\mathcal{O}(\sqrt{d})$ eq. (8), we scale the prior accordingly, e.g., by reparameterizing $\alpha \to \alpha\sqrt{d}$ for the Pareto prior. Moreover, the temporal component of the preconditioner scales as $A_{tt} = \mathcal{O}(d^{-1})$ eq. (83), such that the preconditioned t-drift scales as $\mathcal{O}(d^{-1/2})$. We therefore rescale the noise-variance component of the preconditioner by $\sqrt{d}$, i.e., $\boldsymbol{A} \to \mathrm{diag}(\boldsymbol{I}_d, \sqrt{d})\boldsymbol{A}$, yielding an approximately dimension-independent effective drift. Thus, this has the effect of setting an effective step-size $\eta_\tau = \eta\sqrt{d}$.

The resulting sampling procedure is summarized in algorithm 3.

C.2 Derivation of iDSDS SDE (Equation (13))

C.2.1 State-Dependent Mass Matrix

We outline the derivation from Girolami & Calderhead (2011, Eq. (15)) for a state-dependent preconditioner. Let $\boldsymbol{z}$ denote the state variable and introduce the state-dependent mass matrix $\boldsymbol{M}(\boldsymbol{z}) = \boldsymbol{A}(\boldsymbol{z})^{-1}$ The corresponding lifted target is

$$p(\boldsymbol{z}, \boldsymbol{p}) = p(\boldsymbol{z})\,\mathcal{N}(\boldsymbol{p}; \mathbf{0}, \boldsymbol{M}(\boldsymbol{z}))\,. \tag{92}$$

The state dependence of $\boldsymbol{M}$ leaves the marginal target unchanged, $\int p(\boldsymbol{z}, \boldsymbol{p})\,\mathrm{d}\boldsymbol{p} = p(\boldsymbol{z})$. The associated Hamiltonian is, up to an additive constant,

$$H(\boldsymbol{z}, \boldsymbol{p}) = -\log p(\boldsymbol{z}) + \frac{1}{2}\log\det \boldsymbol{M}(\boldsymbol{z}) + \frac{1}{2}\boldsymbol{p}^\top \boldsymbol{M}(\boldsymbol{z})^{-1}\boldsymbol{p}. \tag{93}$$

Using $\boldsymbol{A}(\boldsymbol{z}) = \boldsymbol{M}(\boldsymbol{z})^{-1}$, Hamilton's equations give

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{z}}{\mathrm{d}s} &= \nabla_{\boldsymbol{p}} H(\boldsymbol{z}, \boldsymbol{p}) = \boldsymbol{A}(\boldsymbol{z})\boldsymbol{p}, \\ \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}s} &= -\nabla_{\boldsymbol{z}} H(\boldsymbol{z}, \boldsymbol{p}) = \boldsymbol{b}(\boldsymbol{z}, \boldsymbol{p}), \end{aligned} \tag{94}$$

where

$$\boldsymbol{b}(\boldsymbol{z}, \boldsymbol{p}) = \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}) + \frac{1}{2}\nabla_{\boldsymbol{z}} \log\det \boldsymbol{A}(\boldsymbol{z}) - \frac{1}{2}\nabla_{\boldsymbol{z}} \left[\boldsymbol{p}^\top \boldsymbol{A}(\boldsymbol{z})\boldsymbol{p}\right]. \tag{95}$$

Adding a Langevin thermostat to the momentum yields the kinetic Langevin dynamics

$$\begin{aligned} \mathrm{d}\boldsymbol{z}_s &= \boldsymbol{A}(\boldsymbol{z}_s)\boldsymbol{p}_s \,\mathrm{d}s, \\ \mathrm{d}\boldsymbol{p}_s &= \boldsymbol{b}(\boldsymbol{z}_s, \boldsymbol{p}_s)\,\mathrm{d}s - \boldsymbol{\Gamma}(\boldsymbol{z}_s)\boldsymbol{A}(\boldsymbol{z}_s)\boldsymbol{p}_s \,\mathrm{d}s + \sqrt{2\boldsymbol{\Gamma}(\boldsymbol{z}_s)}\,\mathrm{d}\boldsymbol{w}_s. \end{aligned} \tag{96}$$

C.2.2 Specular Reflection

Since the noise-variance coordinate is restricted to $t \in [t_{\min}, t_{\max}]$, the kinetic dynamics require a boundary condition for the momentum. We impose specular reflection, the kinetic analogue of reflecting overdamped dynamics (Du et al., 2026, Eq. (4)). At a boundary hitting time s, the position remains continuous while the momentum is mapped as

$$\boldsymbol{p}_{s^+} = \mathcal{R}_{\boldsymbol{z}_s}(\boldsymbol{p}_{s^-}), \tag{97}$$

where $\boldsymbol{p}_{s^-}$ and $\boldsymbol{p}_{s^+}$ denote the momentum immediately before and after reflection, respectively.

For the kinetic energy $K(\boldsymbol{z}, \boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^\top \boldsymbol{A}(\boldsymbol{z})\boldsymbol{p}$, requiring the boundary impulse to act along the unit normal $\boldsymbol{n}$ while preserving K gives the reflection map

$$\mathcal{R}_{\boldsymbol{z}}(\boldsymbol{p}) = \boldsymbol{p} - 2\frac{\boldsymbol{n}^\top \boldsymbol{A}(\boldsymbol{z})\boldsymbol{p}}{\boldsymbol{n}^\top \boldsymbol{A}(\boldsymbol{z})\boldsymbol{n}}\boldsymbol{n}. \tag{98}$$

For iDSDS, since only t is reflected, we have $\boldsymbol{n} = \pm\boldsymbol{e}_t$. Since $\boldsymbol{A}(\boldsymbol{z})$ is block diagonal, eq. (98) reduces to

$$\boldsymbol{p}_x^+ = \boldsymbol{p}_x^-, \qquad p_t^+ = -p_t^-. \tag{99}$$

Under the change of variables $\tau = \log t$, the boundary is mapped to $\tau \in \{\tau_{\min}, \tau_{\max}\}$ and the corresponding momentum satisfies $p_\tau = t\, p_t$ (follows from the definition of the canonical momentum). Since t is unchanged at the instant of reflection, the same sign reversal applies in log-variance.

C.2.3 Discretization

Next, we outline the idea of Falk et al. (2025) to simplify the update rule in eq. (96) by eliminating the momentum variable.

First, we choose the friction matrix proportional to the mass matrix,

$$\boldsymbol{\Gamma}(\boldsymbol{z}) = \varepsilon \boldsymbol{M}(\boldsymbol{z}) = \varepsilon \boldsymbol{A}(\boldsymbol{z})^{-1}, \tag{100}$$

which yields an isotropic, metric-independent momentum relaxation rate (i.e. ε only tunes how inertial the sampler is). Plugging into eq. (96)

$$\begin{aligned} \mathrm{d}\boldsymbol{z} &= \boldsymbol{A}(\boldsymbol{z})\boldsymbol{p}\,\mathrm{d}s, \\ \mathrm{d}\boldsymbol{p} &= [\boldsymbol{b}(\boldsymbol{z}, \boldsymbol{p}) - \varepsilon \boldsymbol{p}]\,\mathrm{d}s + \sqrt{2\varepsilon \boldsymbol{A}(\boldsymbol{z})}\,\mathrm{d}\boldsymbol{w}. \end{aligned} \tag{101}$$

Let $h > 0$ denote the discretization step and define the variables $\beta = 1 - \varepsilon h$ and $\eta = h^2$. Falk et al. (2025) use a single scalar preconditioner $\theta = 1/L$, where θ plays the role of the inverse mass. Apply a Gauss-Seidel-like discretization to update the momentum

$$\boldsymbol{p}_{k+1} = \beta \boldsymbol{p}_k + h\boldsymbol{b}_k + \sqrt{2(1-\beta)\boldsymbol{A}_k}\boldsymbol{\xi}_k, \tag{102}$$

and subsequently update the state using the new momentum,

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + h\boldsymbol{A}_k \boldsymbol{p}_{k+1}, \tag{103}$$

where we use the short-hand notation $\boldsymbol{A}_k = \boldsymbol{A}(\boldsymbol{z}_k)$ and $\boldsymbol{b}_k = \boldsymbol{b}(\boldsymbol{z}_k, \boldsymbol{p}_k)$. The crucial step is identifying the relation

$$\boldsymbol{p}_k = \frac{1}{h}\boldsymbol{A}_{k-1}^{-1}(\boldsymbol{z}_k - \boldsymbol{z}_{k-1}). \tag{104}$$

Substituting this relation into eq. (102) and the resulting expression for $\boldsymbol{p}_{k+1}$ into eq. (103) eliminates the explicit momentum variable and yields

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \beta\boldsymbol{A}_k\boldsymbol{A}_{k-1}^{-1}(\boldsymbol{z}_k - \boldsymbol{z}_{k-1}) + \eta\boldsymbol{A}_k\boldsymbol{b}_k + \sqrt{2\eta(1-\beta)\boldsymbol{A}_k}\boldsymbol{\xi}_k, \tag{105}$$

where the drift is given in eq. (95)

State-only reflection. In the position-only discretization, momentum is represented implicitly through the displacement relation in eq. (104). Therefore, after reflecting an unconstrained proposal $\tilde{t}_{k+1}$ at the boundary using eq. (91), the auxiliary previous position must be updated consistently so that the reconstructed momentum undergoes the same specular reflection.

Let $\sigma(\tilde{t}_{k+1}) \in \{-1, +1\}$ denote the orientation of the reflection, with $\sigma = -1$ for an odd number of boundary reflections and $\sigma = +1$ otherwise. We set

$$t_k^{\text{prev}} = t_{k+1} - \sigma(\tilde{t}_{k+1})\left(\tilde{t}_{k+1} - t_k\right), \tag{106}$$

where t_{k+1} is the reflected position. This ensures that the momentum reconstructed at the next iteration has its normal component reversed, consistent with the specular reflection in eq. (98).

C.2.4 Explicit Terms for the Fisher Preconditioner

The update rule eq. (105) requires three explicit quantities.

Normal noise-variance. For the Fisher preconditioner eq. (83), the transport term is

$$\boldsymbol{A}(\boldsymbol{z}_k)\boldsymbol{A}(\boldsymbol{z}_{k-1})^{-1} = \begin{pmatrix} \frac{t_k}{t_{k-1}}\boldsymbol{I}_d & \boldsymbol{0} \\ \boldsymbol{0}^\top & \frac{a_t(t_k)}{a_t(t_{k-1})} \end{pmatrix}. \tag{107}$$

The force in eq. (95) additionally requires two geometric quantities. First,

$$\nabla_{\boldsymbol{z}} \log\det \boldsymbol{A}(\boldsymbol{z}_k) = \begin{pmatrix} \boldsymbol{0} \\ \frac{d}{t_k} + \frac{\partial_t a_t(t_k)}{a_t(t_k)} \end{pmatrix}, \tag{108}$$

where we used $\det \boldsymbol{A}(\boldsymbol{z}) = t^d\, a_t(t)$. Second

$$\nabla_{\boldsymbol{z}}\left[\boldsymbol{p}_k^\top \boldsymbol{A}(\boldsymbol{z}_k)\boldsymbol{p}_k\right] = \begin{pmatrix} \boldsymbol{0} \\ \|\boldsymbol{p}_{\boldsymbol{x},k}\|^2 + \partial_t a_t(t_k) p_{t,k}^2 \end{pmatrix}, \tag{109}$$

where from the block-diagonal structure of $\boldsymbol{A}$, $\boldsymbol{p}_k^\top \boldsymbol{A}(\boldsymbol{z}_k)\boldsymbol{p}_k = t_k \|\boldsymbol{p}_{\boldsymbol{x},k}\|^2 + a_t(t_k)p_{t,k}^2$,and $\boldsymbol{p}_k = (\boldsymbol{p}_{\boldsymbol{x},k}, p_{t,k})$ is reconstructed from the current and preceeding state eq. (104).

Log variance. The corresponding expressions in log variance follow from the transformation $t = e^\tau$ together with the transformed temporal preconditioner and its derivative. These relations are given in appendix C.1.3.

C.2.5 iDSDS Algorithm

For iDSDS, we apply the position-only update in eq. (105) in log variance $\tau = \log t$ and impose the position-only reflection described above.

Remark on dimensionality scaling. The same dimensionality-scaling argument as for DSDS applies to iDSDS. Accordingly, we scale the noise-variance prior as described above and use the dimensionally rescaled temporal step size $\eta_\tau = \sqrt{d}\,\eta$. In the position-only discretization, η_τ is applied consistently to the temporal momentum reconstruction, drift, and diffusion terms.

The resulting sampler is summarized in algorithm 4.

Algorithm 4: Dual Score Diffusion Sampler with Momentum (iDSDS)

Input: energy model $U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t)$, noise-variance prior $p_{\boldsymbol{\phi},\tau}(\tau)$, initial log-variance $\tau_0 \in [\tau_{\min}, \tau_{\max}]$, number of Langevin steps L, step size η, momentum coefficient $\beta \in [0, 1)$, Fisher preconditioner $\boldsymbol{A}(\tau) = \mathrm{diag}(e^{\tau}\boldsymbol{I}_d, a_\tau(\tau))$ from eq. (86).
Output: sample $\boldsymbol{x}_L$

Set the dimensionally scaled temporal step size $\eta_\tau \leftarrow \sqrt{d}\,\eta$
Initialize $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{0}, t_{\max}\boldsymbol{I}_d)$, $\boldsymbol{z}_0 = (\boldsymbol{x}_0, \tau_0)$, $\tau_{-1} \leftarrow \tau_0$, and $\Delta_0 = \boldsymbol{0}$
for $k = 0$ **to** $L - 1$ **do**
  Reconstruct the implicit momentum $\boldsymbol{p}_k \leftarrow \left(\eta^{-1/2} e^{-\tau_{k-1}} \Delta_{\boldsymbol{x},k},\ \eta_\tau^{-1/2} a_\tau(\tau_{k-1})^{-1} \Delta_{\tau,k}\right)$ eq. (104)
  Compute $\nabla_{\boldsymbol{x}} U_{\boldsymbol{\theta}}(\boldsymbol{x}_k \mid e^{\tau_k})$, $\partial_\tau U_{\boldsymbol{\theta}}(\boldsymbol{x}_k \mid e^{\tau_k})$, and $\partial_\tau \log p_{\boldsymbol{\phi},\tau}(\tau_k)$ eq. (10)
  Compute the Riemannian force $\boldsymbol{b}_k = \boldsymbol{b}(\boldsymbol{z}_k, \boldsymbol{p}_k)$ from eq. (95)
  Sample $\boldsymbol{\xi}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ and $\xi_k \sim \mathcal{N}(0, 1)$
  $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \beta\, e^{\tau_k - \tau_{k-1}} \Delta_{\boldsymbol{x},k} + \eta\, e^{\tau_k} b_{\boldsymbol{x},k} + \sqrt{2\eta(1-\beta)e^{\tau_k}}\, \boldsymbol{\xi}_k$
  $\tilde{\tau}_{k+1} \leftarrow \tau_k + \beta \frac{a_\tau(\tau_k)}{a_\tau(\tau_{k-1})} \Delta_{\tau,k} + \eta_\tau a_\tau(\tau_k) b_{\tau,k} + \sqrt{2\eta_\tau(1-\beta)a_\tau(\tau_k)}\, \xi_k$
  $\tau_{k+1} \leftarrow \mathrm{Reflect}\,(\tilde{\tau}_{k+1}; \tau_{\min}, \tau_{\max})$ eq. (91)
  $\Delta_{k+1} \leftarrow \begin{pmatrix} \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \\ \sigma(\tilde{\tau}_{k+1})(\tilde{\tau}_{k+1} - \tau_k) \end{pmatrix}$ eq. (98)
return $\boldsymbol{x}_L$

C.3 Proof of Sample Readout (Proposition 3.1)

We show that, assuming the joint sampler has reached its invariant distribution $p(\boldsymbol{x}, t)$,

$$\mathcal{W}_2\left(p_{\boldsymbol{x}}, p_{\mathrm{data}}\right) \le \sqrt{d\, \mathbb{E}_{t\sim p(t)}[t]}. \tag{110}$$

Here, $p_{\boldsymbol{x}}(\boldsymbol{x}) = \int_{t_{\min}}^{t_{\max}} p(\boldsymbol{x} \mid t)\, p(t)\, \mathrm{d}t$. describes the distribution of the returned sample $\boldsymbol{x}$ after discarding the noise-variance coordinate.

Proof. We first construct an explicit coupling between $p_{\boldsymbol{x}}$ and p_{data}. Define

$$\pi(\boldsymbol{x}, \boldsymbol{x}_0) = p_{\mathrm{data}}(\boldsymbol{x}_0) \int_{t_{\min}}^{t_{\max}} p(t)\, q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\, \mathrm{d}t, \tag{111}$$

where $q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)$ denotes the Gaussian VE corruption kernel from eq. (15).

The first marginal of π is

$$\begin{aligned} \int \pi(\boldsymbol{x}, \boldsymbol{x}_0)\, \mathrm{d}\boldsymbol{x}_0 &= \int_{t_{\min}}^{t_{\max}} \int q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\, p_{\mathrm{data}}(\boldsymbol{x}_0)\, p(t)\, \mathrm{d}\boldsymbol{x}_0\, \mathrm{d}t \\ &\overset{\text{(i)}}{=} \int_{t_{\min}}^{t_{\max}} p(\boldsymbol{x} \mid t)\, p(t)\, \mathrm{d}t \\ &\overset{\text{(ii)}}{=} p_{\boldsymbol{x}}(\boldsymbol{x}), \end{aligned} \tag{112}$$

where (i) follows from eq. (18), and (ii) uses the definition of $p_{\boldsymbol{x}}$ in appendix C.3

The second marginal is

$$\begin{aligned} \int \pi(\boldsymbol{x}, \boldsymbol{x}_0)\, \mathrm{d}\boldsymbol{x} &= p_{\mathrm{data}}(\boldsymbol{x}_0) \int_{t_{\min}}^{t_{\max}} p(t) \int q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}t \\ &= p_{\mathrm{data}}(\boldsymbol{x}_0), \end{aligned} \tag{113}$$

since $q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)$ is normalized in $\boldsymbol{x}$ and $p(t)$ is a probability density.

Hence, π is an admissible coupling of $p_{\boldsymbol{x}}$ and p_{data}. By the definition of the Wasserstein-2 distance,

$$\begin{aligned}\mathcal{W}_2^2\left(p_{\boldsymbol{x}}, p_{\text{data}}\right) &\overset{\text{(i)}}{\leq} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}_0)\sim\pi}\left[\|\boldsymbol{x}-\boldsymbol{x}_0\|^2\right] \\ &\overset{\text{(ii)}}{=} \mathbb{E}_{\substack{t\sim p(t),\\ \boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}}\left[t\,\|\boldsymbol{\varepsilon}\|^2\right] \\ &\overset{\text{(iii)}}{=} \mathbb{E}_{t\sim p(t)}[t]\,\mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}\left[\|\boldsymbol{\varepsilon}\|^2\right] \\ &\overset{\text{(iv)}}{=} d\,\mathbb{E}_{t\sim p(t)}[t].\end{aligned} \tag{114}$$

Here, (i) follows because π is an admissible coupling, (ii) from the VE reparameterization in eq. (16), (iii) from the independence of t and $\boldsymbol{\varepsilon}$, and (iv) from eq. (20).

Taking the square root yields $\mathcal{W}_2\left(p_{\boldsymbol{x}}, p_{\text{data}}\right) \leq \sqrt{d\,\mathbb{E}_{t\sim p(t)}[t]}$, which proves the claim. □

D Experimental Details

We use random seeds $\{0, 1, 2, 3, 4\}$ for hyperparameter tuning, $\{10, 11, 12, 13, 14\}$ for sample generation, and 42, 43, and 44 for the training, validation, and test splits, respectively.

D.1 Datasets

Gaussian mixture models. We consider K-component Gaussian mixture models with weights w_k, means $\boldsymbol{\mu}_k$ and isotropic standard deviations σ_k

$$p_{\text{data}}(\boldsymbol{x}) = \sum_{k=1}^{K} w_k\,\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \sigma_k^2\,\boldsymbol{I}_d). \tag{115}$$

For the 1D case, we set $K = 3$ with $(w_k) = (0.2, 0.5, 0.3)$, $(\mu_k) = (-4, 0, 3)$, and $(\sigma_k) = (0.5, 1.0, 0.7)$.

For the 2D case, we use $K = 9$ equally weighted components ($w_k = 1/9$) arranged on a 3×3 grid, with $\boldsymbol{\mu}_k \in \{-3, 0, 3\} \times \{-3, 0, 3\}$, and $\sigma_k = 0.5$ for all components.

For training we draw a fixed set of $100{,}000$ samples for the 1D case and $500{,}000$ samples for the 2D case from the respective GMM. For tuning and evaluation we draw a fixed set of $50{,}000$ samples each.

Imaging data. We evaluate on MNIST (Le Cun et al., 1998) ($60{,}000$ train / $10{,}000$ test, 28×28 grayscale images) and CIFAR-10 (Krizhevsky, 2009) ($50{,}000$ train / $10{,}000$ test, 32×32 RGB images). For hyperparameter tuning of the sampling algorithms, we split off the first $10{,}000$ training samples of each dataset to create a validation set. The remaining samples, $50{,}000$ for MNIST and $40{,}000$ for CIFAR-10, are used for training. The respective hold-out test sets are used for FID evaluation.

D.2 Model Architectures

D.2.1 Energy Model

For all experiments that involve a parametrized score-model we use an energy model of the form

$$U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t) = f_{\boldsymbol{\theta}_f}(\boldsymbol{x}, t)^\top\; w_{\boldsymbol{\theta}_w}(t) + g_{\boldsymbol{\theta}_g}(\boldsymbol{x}), \tag{116}$$

where $f_{\boldsymbol{\theta}_f} : \mathbb{R}^d \times \mathcal{T} \to \mathbb{R}^{d_f}$ is a noise-variance-conditioned backbone model, providing a d_f-dimensional feature vector, $w_{\boldsymbol{\theta}_w} : \mathcal{T} \to \mathbb{R}^{d_f}$ is a noise-variance-dependent weighting on the produced feature vector, and $g_{\boldsymbol{\theta}_g} : \mathbb{R}^d \times \mathcal{T} \to \mathbb{R}$ is a Gaussian correction term, providing scores outside the data-domain based on the prior assumption of an overall quadratic energy landscape. We subsequently refer to the combination of the noise-variance-dependent weighting and the Gaussian correction term as the *energy head*. This formulation is motivated by combining of noise-conditional score-models (Song & Ermon, 2019) with explicitly learned energy potentials (Roth & Black, 2009; Kobler et al., 2020; 2021). Note this energy model differs from the model suggested by Guth et al. (2025) to avoid scaling issues for scores around $\boldsymbol{x} = \mathbf{0}$.

Noise-variance-dependent weighting. The noise-variance-dependent weighting consists of a noise-variance embedding $e : \mathcal{T} \to \mathbb{R}^{d_e}$ followed by a L-layer MLP, where $L = 1$ for GMM and MNIST, $L = 2$ with hidden dimension d_f and GELU non-linearity for CIFAR-10. In practice we share the noise-variance embedding with the backbone model, as we found this to improve training stability and performance.

Gaussian correction term. The Gaussian correction term is of the form

$$g_{\boldsymbol{\theta}_g}(\boldsymbol{x}, t) = \frac{1}{2}\left\|\boldsymbol{x} - \hat{\boldsymbol{\mu}}\right\|^2_{\tilde{\boldsymbol{\Sigma}}(t)^{-1}} \tag{117}$$

where $\tilde{\boldsymbol{\Sigma}}(t) = \mathrm{diag}(\mathrm{softplus}(\hat{\boldsymbol{v}})) + t\boldsymbol{I}_d$. The parameters $\boldsymbol{\theta}_g = \{\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{v}} \in \mathbb{R}^d\}$ are first initialized by the element-wise mean and variance over the dataset and subsequently learned during training. Note that we use the notation $\hat{\boldsymbol{v}}$ to indicate the learned variance term, rather than $\hat{\boldsymbol{\Sigma}}^2$, since the parameter itself is not constrained but gets cast into $\mathbb{R}^+$ via the $\mathrm{softplus}$ function.

This term provides a quadratic prior for scores at the tails of the distribution and is motivated by the Gaussian diffusion process. The space affected by this term corresponds to samples $(\boldsymbol{x}, t)$ where the noise level of $\boldsymbol{x}$ does not match t, and is therefore not usually sampled during training and also not visited during inference, but we found DSDS to benefit from this additional "push towards the dataset mean".

D.2.2 Backbone Model

Vector valued data. For all GMM experiments, we use a noise-conditioned residual feed forward neural network (MLP ResNet). Given an input $\boldsymbol{x} \in \mathbb{R}^d$ and a noise variance $t \in \mathcal{T}$, the network first maps $\boldsymbol{x}$ to a hidden representation in $\mathbb{R}^{d_h}$ via $\boldsymbol{h}_0 = \boldsymbol{W}_0\boldsymbol{x} + \boldsymbol{b}_0$, then passes it through N residual blocks, and finally maps it from $\mathbb{R}^{d_h}$ to $\mathbb{R}^{d_f}$ via $\boldsymbol{h}_f = \boldsymbol{W}_f\boldsymbol{h}_N + \boldsymbol{b}_f$, with $\boldsymbol{W}_0 \in \mathbb{R}^{d_h \times d}$, $\boldsymbol{W}_f \in \mathbb{R}^{d_f \times d_h}$, $\boldsymbol{b}_0 \in \mathbb{R}^{d_h}$, and $\boldsymbol{b}_f \in \mathbb{R}^{d_f}$ respectively.

Each residual block can be described as

$$\boldsymbol{h}_{k+1} = \boldsymbol{h}_k + \boldsymbol{W}_{k,2}\,\mathrm{Cond}_{k,2}\left[\phi\left(\mathrm{Cond}_{k,1}\left[\mathrm{Norm}_{k,2}\left(\boldsymbol{W}_{k,1}\,\phi\left(\mathrm{Norm}_{k,1}\left(\boldsymbol{h}_k\right)\right), e(t)\right)\right]\right), e(t)\right], \tag{118}$$

with

- $\boldsymbol{W}_{k,1}, \boldsymbol{W}_{k,2} \in \mathbb{R}^{d_h \times d_h}$ bias-free weights,
- $e : \mathcal{T} \to \mathbb{R}^{d_e}$ three-layer MLP noise encoding with hidden dimension $\mathbb{R}^{d_e}$ and SiLU non-linearity applied to $\log(t + t_{\min})$,
- $\mathrm{Cond}_{k,i} : \mathbb{R}^{d_h} \times \mathbb{R}^{d_e} \to \mathbb{R}^{d_h}$ channel-wise noise-conditioning which internally applies a learned projection from $e(t) \in \mathbb{R}^{d_e}$ to $e_{k,i}(t) \in \mathbb{R}^{d_h}$ and conditions each channel as $\boldsymbol{h}_k^{(c)} = \boldsymbol{h}_k^{(c)}\mathrm{softplus}(e_{k,i}(t)^{(c)})/\log(2)$,
- ϕ being GELU element-wise non-linearity,
- and $\mathrm{Norm}_{k,i}$ being identity in this specific networks, i.e., no normalization.

We specifically use $N = 4$, $d_h = 32$, $d_f = 16$ $d_e = 16$, yielding a total trainable parameter count of 13,712 for the backbone network. The full energy model, including the noise-dependent weighting and the Gaussian correction term, has 13,986 trainable parameters.

Imaging data. For imaging data (MNIST, CIFAR-10), we use a modified version of the NCSN++ U-net architecture of Song et al. (2021), based on the implementation by Karras et al. (2022)[2]. The particular model hyperparameters can be seen in table 3.

The changes we made are:

- For greater expressivity, we use pre- and post-activation conditioning rather than pre-activation only. The convolutional block structure therefore largely mirrors that of MLP ResNet in eq. (118), with group norm layers, skip weighting, SiLU activation, and additive conditioning to match that of NCSN++.

[2] https://github.com/nvlabs/edm

Table 3: Backbone hyperparameters and trainable parameter counts for the MNIST and CIFAR-10 models. *Out channels* corresponds to d_f. *Base model channels* denotes the channel width at the coarsest resolution level; channel widths at deeper levels are obtained by multiplying this base width by the corresponding entry of *channel multipliers*. *Blocks per resolution* is the number of residual blocks applied at each resolution level, in both the encoder and the decoder. *Resample filter* specifies the FIR filter kernel used for the up-/downsampling convolutions. Trainable parameters are reported both for the backbone alone and for the total energy model, including the noise-dependent weighting and the Gaussian correction term.

	MNIST	**CIFAR-10**
Input resolution	28	32
In / out channels	1 / 16	3 / 32
Base model channels	32	128
Channel multipliers	[1, 2]	[2, 2, 2]
Blocks per resolution	2	3
Dropout	0.13	0.13
Resample filter	[1, 1]	[1, 3, 3, 1]
# Trainable parameters backbone	1,000,160	52,570,720
# Trainable parameters total	1,003,792	52,586,144

- We do not use an attention mechanism, as we found it to cause significant memory and stability issues during training.
- We do not use class-label conditioning or augmentation-label conditioning.

In order to produce a feature vector for the energy-head, we use a higher number of output channels d_f and sum over all spatial dimension, i.e., the network output shape is `(B,` d_f`,H,W)`, which we collapse into `(B,` d_f`)`.

D.3 Training Details

D.3.1 Dual score matching loss

We train our model using dual score matching loss $\mathcal{L}_{\mathrm{ddsm}} = \mathcal{L}_{\mathrm{dsm}} + \mathcal{L}_{\mathrm{tsm}}$ (Kobler & Pock, 2023; Guth et al., 2025). Here, the denoising score-matching (Song & Ermon, 2019) and the temporal score-matching terms are

$$\begin{aligned}
\mathcal{L}_{\mathrm{dsm}}(\theta) &= \mathbb{E}_{\substack{\log t \sim \mathcal{U}[\log t_{\min}, \log t_{\max}] \\ \boldsymbol{x}_0 \sim p_{\mathrm{data}} \\ \boldsymbol{x} \sim q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)}} \left[\lambda_{\mathrm{dsm}}(t) \left\| \nabla_{\boldsymbol{x}} U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t) + \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) \right\|^2 \right] \\
\mathcal{L}_{\mathrm{tsm}}(\theta) &= \mathbb{E}_{\substack{\log t \sim \mathcal{U}[\log t_{\min}, \log t_{\max}] \\ \boldsymbol{x}_0 \sim p_{\mathrm{data}} \\ \boldsymbol{x} \sim q(\boldsymbol{x} \mid \boldsymbol{x}_0, t)}} \left[\lambda_{\mathrm{tsm}}(t) \left\| \partial_t U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t) + \partial_t \log q(\boldsymbol{x} \mid \boldsymbol{x}_0, t) \right\|^2 \right].
\end{aligned} \tag{119}$$

Both target scores are available in closed form from eq. (17). Following Guth et al. (2025), we choose $\lambda_{\mathrm{dsm}}(t) = t/d$ and $\lambda_{\mathrm{tsm}}(t) = (t/d)^2$.

D.3.2 Training Hyperparameters

The relevant training hyperparameters can be seen in table 4.

D.3.3 Limitations

Double back-propagation and memory issues. We calculate $\nabla_{\boldsymbol{x}} U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t)$ and $\partial_t U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t)$ via PyTorch automatic differentiation, meaning that during training we use double back-propagation: first to obtain the scores $\nabla_{\boldsymbol{x}} U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t)$ and $\partial_t U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t)$, then to obtain the gradients of the parameters $\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta})$. The need for double back-propagation increases memory consumption, training time, and inference time, as the scores are not acquired via a single forward pass, but rather a forward pass followed by a backward pass.

Table 4: Training hyperparameters for the GMM, MNIST, and CIFAR-10 experiments. *Effective batch size* is the product of # GPUs, # gradient accumulation steps, and the per-step per-GPU batch size. *Msamp* denotes millions of training samples. The learning rate is linearly warmed up from (start factor $\times \mathrm{lr}_0$) to lr_0 over the warmup duration, then follows the given schedule down to the final LR lr_f over the remaining training duration. *Var. scale* is the factor by which the initial variance of the Gaussian correction term is scaled.

	GMM	**MNIST**	**CIFAR-10**
GPU type	Titan V	Titan V	A100 (MIG)
GPU memory [GB]	12	12	20
# GPUs	1	2	2
# Gradient accumulation steps	1	1	5
Batch size [per grad. acc. step per GPU]	512	256	48
Effective batch size	512	512	480
Total training duration [Msamp]	2	50	20
# Updates	3,901	99,610	42,708
Total training time [hours]	0.015	12	96
Training set size	100,000	50,000	40,000
Validation set size	50,000	10,000	10,000
Data augmentation	–	–	50% horizontal flip
Var. scale	0.25	1	1
Optimizer	Adam	Adam	AdamW
Gradient clipping [max norm]	1	1	1
Base LR (lr_0)	10^{-3}	10^{-3}	10^{-4}
LR warmup start factor	–	10^{-8}	10^{-8}
LR warmup duration [Msamp]	0	1	0.5
LR schedule (post-warmup)	cosine	constant	linear
Final LR (lr_f)	10^{-6}	10^{-3}	10^{-8}

D.4 Sampling Algorithms

D.4.1 Langevin Baselines

ULA. The Unadjusted Langevin Algorithm (ULA) is obtained by applying the Euler–Maruyama discretization to eq. (2).

ILA. The Inertial Langevin Algorithm (ILA) of Falk et al. (2025) uses a position-only discretization of underdamped Langevin dynamics eq. (4) by eliminating the explicit momentum variable. We use the same discretization principle for iDSDS, and details are described in appendix C.2.3.

ALD. The original formulation of ALD by Song & Ermon (2019, Algorithm. 1) uses a noise-dependent step size, scaled proportionally to the noise variance $\alpha_k = \epsilon\, \sigma_k^2/\sigma_{\min}^2$. In our formulation, this dependence is instead absorbed into the preconditioner $\boldsymbol{A}(t) = t\boldsymbol{I}_d$, allowing us to use a constant step size η. Accounting additionally for the factor-of-two difference between their Langevin convention and eq. (2), we set $\eta \coloneqq \epsilon/(2t_{\min})$. With this choice, the update is equivalent to the original ALD formulation, identifying their squared noise scale σ^2 with our variance parameter t.

D.4.2 Reverse SDE Solvers

Score-based diffusion sampling can equivalently be formulated through stochastic differential equations (SDEs) (Song et al., 2021). For the VE Gaussian perturbation process in eq. (15), the corresponding forward SDE is

$$\mathrm{d}\boldsymbol{x}_t = \mathrm{d}\boldsymbol{w}_t, \qquad t \in [0, T]. \tag{120}$$

Under suitable regularity conditions, the diffusion admits the reverse-time SDE (Anderson, 1982),

$$\mathrm{d}\boldsymbol{x}_t = -\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)\, \mathrm{d}t + \mathrm{d}\bar{\boldsymbol{w}}_t, \qquad t \in [T, 0], \tag{121}$$

where $\bar{\boldsymbol{w}}_t$ denotes a reverse-time Wiener process.

EM. We numerically integrate eq. (121) from T toward zero using noise levels $\{t_k\}_{k=1}^K$ with $t_1 = T > \cdots > t_K \approx 0$ and the Euler–Maruyama (EM) discretization. Details see Song et al. (2021, Algorithm. 1)

PC. The predictor-corrector (PC) sampler of Song et al. (2021, Algorithm. 2) augments each reverse-SDE predictor step with one or more score-based MCMC corrector steps. We use Langevin dynamics as the corrector at the updated noise level. While Song et al. (2021) choose the corrector step size adaptively from a prescribed signal-to-noise ratio (Appendix G), we instead use the same preconditioned Langevin update as in our ALD baseline with a constant stepsize η.

D.4.3 Additional DSDS variants

LG. In their work, Kim & Ye (2023, Sec. 4) use Gibbs sampling to update the joint state $(\boldsymbol{x}, t)$. Concretely, they update $\boldsymbol{x}$ via a Langevin step targeting $p(\boldsymbol{x} \mid t)$ and update t by approximately sampling from the conditional $p(t \mid \boldsymbol{x})$. To this end, the interval $\mathcal{T} = [t_{\min}, t_{\max}]$ is discretized into J bins with representative noise levels $\{t_j\}_{j=1}^J$. The corresponding categorical probabilities are $\pi_j(\boldsymbol{x}) \propto p(\boldsymbol{x}, t_j)\,\Delta t_j$, where Δt_j denotes the width of the corresponding bin in physical noise space. Sampling from the resulting categorical distribution determines the subsequent noise level.

While Kim & Ye (2023) train a noise-level classifier to approximate $p(t \mid \boldsymbol{x})$, our energy formulation (combined with dual score matching) allows us to evaluate the joint density up to a global normalization constant as $p(\boldsymbol{x}, t) \propto \exp(-U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t))p(t)$. Consequently, the conditional $p(t \mid \boldsymbol{x})$ can be approximated directly on the discrete noise grid, with categorical logits $\ell_j = -U_{\boldsymbol{\theta}}(\boldsymbol{x} \mid t_j) + \log p(t_j) + \log \Delta t_j$. A drawback of this approach is that each Gibbs update requires evaluating the joint density at all J candidate noise levels, whereas DSDS only requires local score evaluations at the current state.

D.5 Noise-Variance Schedules

We consider three noise-variance schedules for the baseline samplers (ALD, EM, PC). Let N denote the number of noise levels and $r_i = i/(N-1)$ for $i = 0, \ldots, N-1$. The schedules are given by

$$t_i = \begin{cases} t_{\max} + r_i\,(t_{\min} - t_{\max})\,, & \text{Linear,} \\ t_{\max}\left(\dfrac{t_{\min}}{t_{\max}}\right)^{r_i}, & \text{Geometric,} \\ \left[t_{\max}^{1/(2\rho)} + r_i\left(t_{\min}^{1/(2\rho)} - t_{\max}^{1/(2\rho)}\right)\right]^{2\rho}, & \text{EDM.} \end{cases} \tag{122}$$

The geometric schedule is commonly used for VE diffusion models (Song & Ermon, 2019), while the EDM schedule is proposed by Karras et al. (2022), with ρ controlling the distribution of noise levels.

D.6 Prior Distributions

This section gives an overview of the prior distriubtions investigated within this work. Note that we define the PDFs $p(t)$ on their natural supports rather than truncating them to $\mathcal{T} = [t_{\min}, t_{\max}]$. The reflecting boundaries restrict the stationary density to $p(t)\mathbf{1}_{\mathcal{T}}(t)$ up to normalization, which leaves the log-density derivatives used in the sampling dynamics unchanged within $\mathcal{T}$.

Exponential distribution. It has support $t \in [0, \infty)$ and rate parameter $\lambda > 0$, with

$$\begin{aligned} p(t) &= \lambda e^{-\lambda t}, \\ \partial_t \log p(t) &= -\lambda, \\ \partial_t^2 \log p(t) &= 0. \end{aligned} \tag{123}$$

Gamma distribution. It has support $t \in (0, \infty)$, with shape parameter $\alpha > 0$ and rate parameter $\beta > 0$, with

$$\begin{aligned} p(t) &= \frac{\beta^\alpha}{\Gamma(\alpha)} t^{\alpha-1} e^{-\beta t}, \\ \partial_t \log p(t) &= \frac{\alpha - 1}{t} - \beta, \\ \partial_t^2 \log p(t) &= -\frac{\alpha - 1}{t^2}. \end{aligned} \tag{124}$$

Half-normal distribution. It has support $t \in [0, \infty)$, with scale parameter $\sigma > 0$, with

$$\begin{aligned} p(t) &= \frac{\sqrt{2}}{\sigma\sqrt{\pi}} e^{-\frac{t^2}{2\sigma^2}}, \\ \partial_t \log p(t) &= -\frac{t}{\sigma^2}, \\ \partial_t^2 \log p(t) &= -\frac{1}{\sigma^2}. \end{aligned} \tag{125}$$

Half-Student-t distribution. It has support $t \in [0, \infty)$, with scale parameter $\sigma > 0$ and degrees of freedom $\nu > 0$, with

$$\begin{aligned} p(t) &= \frac{2\,\Gamma\left(\frac{\nu+1}{2}\right)}{\sigma\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{t^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}}, \\ \partial_t \log p(t) &= -\frac{(\nu+1)t}{t^2 + \nu\sigma^2}, \\ \partial_t^2 \log p(t) &= \frac{(\nu+1)(t^2 - \nu\sigma^2)}{(t^2 + \nu\sigma^2)^2}. \end{aligned} \tag{126}$$

Log-normal distribution. It has support $t \in (0, \infty)$, with location parameter $\mu \in \mathbb{R}$ and scale parameter $\sigma > 0$, with

$$\begin{aligned} p(t) &= \frac{1}{t\sigma\sqrt{2\pi}} \exp\left(-\frac{(\log t - \mu)^2}{2\sigma^2}\right), \\ \partial_t \log p(t) &= -\frac{\log t - \mu + \sigma^2}{t\sigma^2}, \\ \partial_t^2 \log p(t) &= \frac{\log t - \mu + \sigma^2 - 1}{t^2\sigma^2}. \end{aligned} \tag{127}$$

Log-uniform distribution. It has support $t \in [t_{min}, t_{max}]$, with $0 < t_{min} < t_{max}$, with

$$\begin{aligned} p(t) &= \frac{1}{t \log(t_{max}/t_{min})}, \\ \partial_t \log p(t) &= -\frac{1}{t}, \\ \partial_t^2 \log p(t) &= \frac{1}{t^2}. \end{aligned} \tag{128}$$

Pareto distribution. It has support $t \in [t_{\min}, \infty)$, with shape parameter $\alpha > 0$ and minimum value $t_{\min} > 0$, with

$$\begin{aligned} p(t) &= \frac{\alpha\, t_{\min}^\alpha}{t^{\alpha+1}}, \\ \partial_t \log p(t) &= -\frac{\alpha + 1}{t}, \\ \partial_t^2 \log p(t) &= \frac{\alpha + 1}{t^2}. \end{aligned} \tag{129}$$

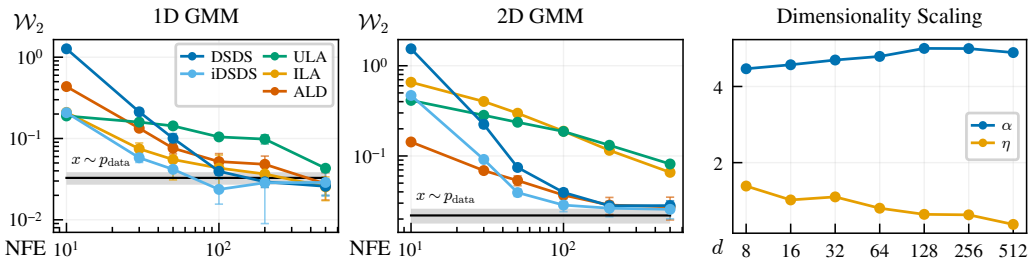


Figure 6: GMM Convergence curves and dimensionality scaling.

Power-exponential distribution. It has support $t \in [0, \infty)$, with rate parameter $\lambda > 0$ and shape parameter $q > 0$, with

$$\begin{aligned} p(t) &= \frac{q\lambda^{1/q}}{\Gamma(1/q)} e^{-\lambda t^q}, \\ \partial_t \log p(t) &= -\lambda q t^{q-1}, \\ \partial_t^2 \log p(t) &= -\lambda q(q-1) t^{q-2}. \end{aligned} \tag{130}$$

Scaled Beta distribution. It has support $t \in [t_{\min}, t_{\max}]$, with shape parameters $\alpha, \beta > 0$. Defining $u = t - t_{\min}/(t_{\max} - t_{\min})$, its density and derivatives are

$$\begin{aligned} p(t) &= \frac{u^{\alpha-1}(1-u)^{\beta-1}}{(t_{\max} - t_{\min}) B(\alpha, \beta)}, \\ \partial_t \log p(t) &= \frac{\alpha - 1}{t - t_{\min}} - \frac{\beta - 1}{t_{\max} - t}, \\ \partial_t^2 \log p(t) &= -\frac{\alpha - 1}{(t - t_{\min})^2} - \frac{\beta - 1}{(t_{\max} - t)^2}. \end{aligned} \tag{131}$$

E Additional Experiments - DSDS

E.1 GMM Convergence Curves

Figure 6 shows the final $\mathcal{W}_2$ distance across NFE budgets for the 1D GMM (left) and 2D GMM (right). iDSDS reduces the error particularly quickly at small budgets, while ALD is also highly competitive in the low-NFE regime, especially in 2D. With increasing NFE, however, DSDS and iDSDS continue to improve and ultimately reach lower error, closer to the finite-sample reference level. The non-annealed ULA and ILA baselines converge more slowly, particularly in 2D. Since improvements beyond roughly 200 evaluations are small, we use $\text{NFE} = 200$ for the main GMM experiments.

E.2 Scaling across dimensions.

To evaluate the proposed dimensionality scaling, we consider the GMM in eq. (115) with $K = 8$, uniform weights $w_k = 1/K$, and fixed isotropic variance $\sigma_k = \sigma = 0.3$. The means form a regular simplex on the unit sphere, satisfying $\|\boldsymbol{\mu}_k\| = 1$ and $\boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j = -1/(K-1)$ for $i \neq j$. Since the simplex spans only $K - 1 = 7$ dimensions, it is embedded into $\mathbb{R}^d$ and randomly rotated to avoid alignment with the coordinate axes. We vary $d \in \{8, 16, 32, 64, 128, 256, 512\}$ while keeping the number of modes, their pairwise separation, and the component variance fixed.

For each dimension, we independently tune DSDS at the same fixed NFE budget and report the optimal parameters after applying the proposed dimension-dependent reparameterization. As shown in fig. 6 (right), α varies only mildly, while the stepsize η remains on a similar scale despite the 64-fold increase in dimension. Since η controls both the spatial and temporal updates, a slight variation of its optimum across dimensions is expected.

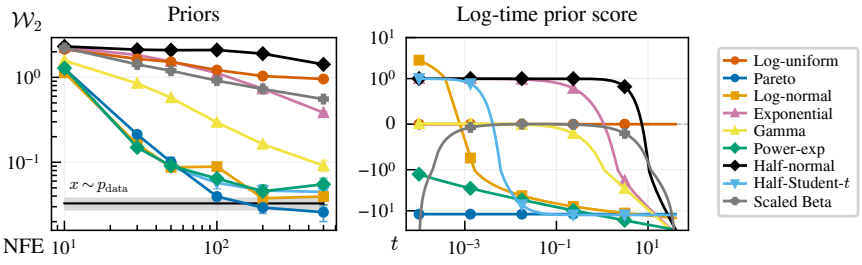


Figure 7: Additional prior distributions.

E.3 Detailed Discussion of Prior Distributions

We consider the prior families defined in appendix D.6. Their behavior becomes particularly transparent in log-time $\tau = \log t$: log-uniform and Pareto priors induce constant scores, the log-normal prior induces a linear score, while exponential and Gamma priors yield scores of the form $c_0 - c_1 e^{\tau}$. Power-exponential and half-normal priors introduce stronger $e^{q\tau}$-type dependence, the half-Student-t prior yields a saturating nonlinear score, and the scaled Beta prior exhibits strong boundary dependence. The resulting log-time scores are shown in fig. 7 (right).

The corresponding sampling performance is shown in fig. 7 (left). Pareto, log-normal, power-exponential, and half-Student-t priors achieve the lowest final $\mathcal{W}_2$ errors, whereas several of the more strongly varying or boundary-sensitive scores converge more slowly. These results indicate that DSDS is not tied to a particular prior family, but that the shape of the log-time score strongly influences how efficiently the sampler traverses the noise dimension. For sampling, we therefore recommend the Pareto prior, which is particularly convenient in log-time: its constant score, combined with Fisher preconditioning, yields a simple state-independent drift in the noise direction.

E.4 Preconditioning

E.4.1 Conditional Fisher Metric

The joint Fisher preconditioner includes both the conditional and prior contributions to the $\tau\tau$ metric. For the Pareto prior, this leads, after dimensional rescaling, to the effective prior drift $\boldsymbol{A}^{\text{joint}}_{\tau\tau}\, \partial_\tau \log p(\tau) = -\alpha/(1/2 + \alpha^2)$. Its magnitude is maximal at $\alpha = \sqrt{1/2}$ and decreases for larger α. Thus, although increasing α strengthens the prior's pull toward small noise levels, this does not translate into an increasingly strong drift under the joint Fisher metric.

We therefore also consider a conditional Fisher variant that omits the prior contribution from the metric in eq. (9). In this case, the effective Pareto drift becomes -2α, such that increasing α directly strengthens the motion toward $\tau_{\min}$ and can in principle accelerate traversal of the noise range.

For the Pareto prior, the conditional and joint mobilities differ only by a constant factor, $\boldsymbol{A}^{\text{joint}}_{\tau\tau} = \boldsymbol{A}^{\text{cond}}_{\tau\tau}/(1+\alpha^2)$. In practice, however, the larger τ-mobility of the conditional metric makes the use of a single shared stepsize η for the $\boldsymbol{x}$- and τ-updates restrictive: stability of the τ-update requires a smaller η, which in turn slows the spatial dynamics and leads to slower overall convergence. Using seperate stepsizes η_x, η_t yields no improvement; see fig. 8 (left). For priors with non-constant log-time scores, the relation between the conditional and joint metrics is no longer a simple constant rescaling; investigating this regime is left for future work.

E.4.2 Effect of Preconditioner-Specific Hyperparameter Tuning

The apparent advantage of Euclidean preconditioning at small NFE is closely tied to its optimal hyperparameters. For a Pareto prior, the Euclidean metric leaves the prior drift proportional to $-\alpha$, such that larger values of α directly induce a stronger pull toward small noise levels. Accordingly, the Euclidean variant selects comparatively large α at low NFE budgets and can traverse the noise range rapidly; the corresponding optimal hyperparameters for Euclidean and Fisher preconditioning are shown in fig. 8 (right).

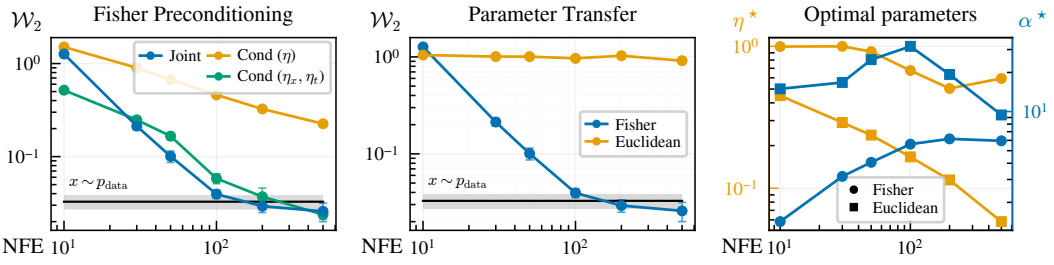


Figure 8: Additional preconditioning study.

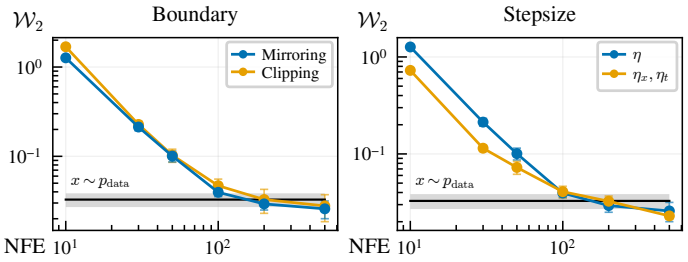


Figure 9: Additional DSDS ablation study with 1D GMM.

To isolate the effect of the preconditioner from that of hyperparameter tuning, we additionally evaluate the Euclidean sampler using the hyperparameters optimized for the Fisher-preconditioned variant. As shown in fig. 8 (middle), performance then degrades substantially and convergence becomes much slower. This parameter-transfer experiment therefore highlights that the Fisher preconditioner provides a genuine acceleration beyond merely inducing a different optimal hyperparameter regime, while adding essentially no computational overhead due to its closed-form expression.

E.5 Additional Ablations

Boundary handling. We investigate the effect of the boundary treatment for the diffusion-time variable by comparing mirroring and clipping. Both yield nearly identical performance across all NFEs (fig. 9 left), indicating that DSDS is largely insensitive to the specific boundary handling in this setting.

Separate step sizes. We additionally allow different step sizes for the spatial and temporal coordinates by scaling the mobility as $\tilde{\boldsymbol{A}}(\boldsymbol{z}) = \boldsymbol{D}_\eta \boldsymbol{A}(\boldsymbol{z})$, with $\boldsymbol{D}_\eta = \mathrm{diag}((\eta_x/\eta)\boldsymbol{I}_d, \eta_t/\eta)$. Optimizing η_x and η_t separately yields only marginal improvements over a shared step size (fig. 9 middle), suggesting that the preconditioner already balances the two update directions effectively.

E.6 Hyperparameter Sensitivity

We assess the sensitivity of DSDS and iDSDS on the 1D GMM by varying one hyperparameter at a time while fixing all remaining parameters to their tuned values. For the stepsize η and Pareto shape parameter α, we report performance as a function of the relative perturbation to the tuned optimum, $\eta/\eta^\star$ and $\alpha/\alpha^\star$. For iDSDS, we additionally vary the momentum parameter β directly.

As shown in fig. 10, both methods are robust to moderate perturbations around their tuned settings. The selected values lie close to the best-performing region in all cases. iDSDS is particularly insensitive to the stepsize, whereas DSDS becomes unstable for a large increase in η. The momentum sweep exhibits a clear optimum, with performance degrading smoothly away from the tuned value.

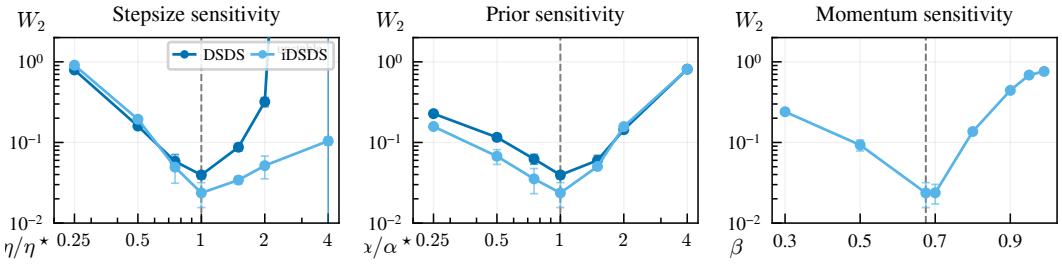


Figure 10: Hyperparameter sensitivity.

Table 5: Results of DSM and DDSM training on the 1D GMM. Final $\mathcal{W}_2$ distance $[\times 10^{-2}]$ ($\downarrow$) is reported as mean $\pm$ standard deviation over five seeds. GT denotes the corresponding sampler using the exact ground-truth score instead of a learned score. Bold indicates the better result between DSM and DDSM.

NFE	Sampler	DSM	DDSM	GT
100	ALD	9.24 ± 1.56	$\mathbf{8.00} \pm 0.35$	4.89 ± 1.29
	EM	$\mathbf{6.84} \pm 0.52$	8.69 ± 0.94	4.85 ± 0.87
	PC	7.12 ± 0.70	$\mathbf{6.25} \pm 0.40$	4.89 ± 1.66
200	ALD	7.97 ± 1.33	$\mathbf{7.59} \pm 0.53$	3.58 ± 1.06
	EM	$\mathbf{6.80} \pm 0.74$	7.68 ± 1.09	3.96 ± 1.09
	PC	6.66 ± 1.19	$\mathbf{6.36} \pm 1.27$	4.18 ± 2.74
500	ALD	7.91 ± 1.62	$\mathbf{7.52} \pm 0.33$	2.59 ± 0.81
	EM	6.78 ± 1.16	$\mathbf{6.63} \pm 0.44$	3.01 ± 0.87
	PC	6.18 ± 1.12	$\mathbf{5.94} \pm 0.33$	3.65 ± 2.12

F Additional Experiments - Baseline Samplers

F.1 Effect of the TSM Objective on Schedule Samplers

We compare DSM- and DDSM-trained models (detail see appendix D.3) for the ALD, EM, and PC samplers across three NFE budgets using the EDM schedule, with results reported in table 5. DDSM achieves lower mean $\mathcal{W}_2$ in seven of the nine settings and often exhibits lower variability across random seeds. We therefore use DDSM-trained models throughout the remainder of the paper and discard the time score for baseline samplers that do not require it.

F.2 Choice of Noise Schedule

Karras et al. (2022) recommend an EDM noise schedule with $\rho = 7$ for their standard-deviation parameterization. Since we instead parameterize the diffusion process by the variance $t = \sigma^2$, we conduct a short study of different noise schedules in fig. 11. We find that $\rho = 7$ also performs consistently well under our parameterization and therefore use it throughout our experiments unless stated otherwise.

G Additional Image Generation Results

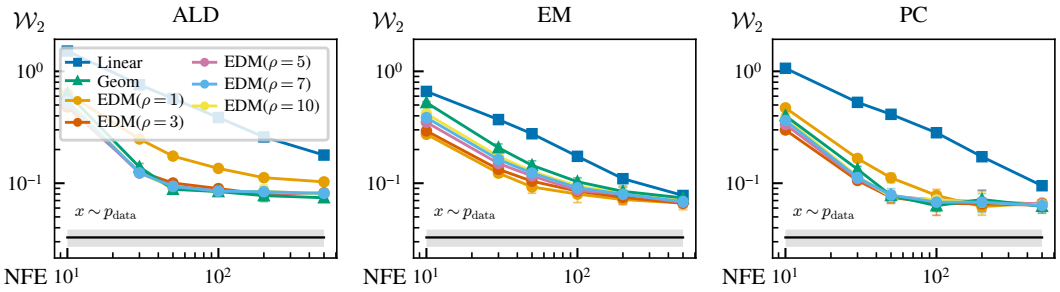


Figure 11: Effect of noise schedule on baseline samplers.

Table 6: FID (↓) on MNIST (top) and CIFAR-10 (bottom), computed from 10,000 generated samples. Results are reported as mean ± standard deviation over five / three seeds. Best results are bolded; second-best results are underlined.

Method	NFE 50	NFE 100	NFE 200	NFE 500	NFE 1000
$\boldsymbol{x} \sim p_{\text{data}}$			1.54 ± 0.02		
ALD	129.17 ± 0.62	58.35 ± 0.79	29.75 ± 0.38	18.85 ± 0.20	15.31 ± 0.33
EM	$\underline{21.65} \pm 0.21$	$\underline{13.38} \pm 0.24$	11.02 ± 0.10	9.98 ± 0.11	$\underline{9.79} \pm 0.09$
PC	25.74 ± 0.22	11.86 ± 0.10	$\mathbf{8.48} \pm 0.16$	$\mathbf{7.10} \pm 0.09$	$\mathbf{6.74} \pm 0.14$
DSDS	316.52 ± 0.32	310.18 ± 0.44	211.23 ± 0.64	45.19 ± 0.12	14.68 ± 0.18
iDSDS	149.11 ± 0.21	$\mathbf{9.36} \pm 0.14$	$\underline{9.72} \pm 0.20$	$\underline{8.99} \pm 0.15$	11.05 ± 0.08

Method	NFE 100	NFE 200	NFE 500	NFE 1000	NFE 2000
$\boldsymbol{x} \sim p_{\text{data}}$			5.19 ± 0.04		
ALD	240.07 ± 0.47	198.96 ± 0.35	144.43 ± 0.89	101.37 ± 0.61	67.72 ± 0.20
EM	$\mathbf{102.00} \pm 0.77$	$\mathbf{61.60} \pm 0.26$	$\underline{46.68} \pm 0.28$	$\underline{42.99} \pm 0.23$	$\underline{41.85} \pm 0.52$
PC	119.45 ± 0.21	$\underline{61.65} \pm 0.30$	$\mathbf{38.40} \pm 0.42$	$\mathbf{31.22} \pm 0.24$	$\mathbf{28.58} \pm 0.31$
DSDS	393.45 ± 0.28	384.24 ± 0.11	341.16 ± 0.14	98.20 ± 0.70	68.67 ± 0.77
iDSDS	$\underline{102.05} \pm 0.28$	93.48 ± 0.26	82.43 ± 0.38	81.98 ± 0.45	94.05 ± 0.82

Table 7: Negative log-likelihood (NLL, ↓) on MNIST (top) and CIFAR-10 (bottom), computed from 10,000 generated samples and evaluated under the trained energy model at $t_{\min}$. Entries report the mean and standard deviation in bits per dimension. Best results are bolded; second-best results are underlined.

Method	NFE 50	NFE 100	NFE 200	NFE 500	NFE 1000
ALD	3.86 ± 0.68	3.13 ± 0.33	2.69 ± 0.30	2.49 ± 0.23	2.38 ± 0.21
EM	$\mathbf{2.90} \pm 0.25$	$\underline{2.53} \pm 0.23$	2.38 ± 0.22	2.31 ± 0.21	2.29 ± 0.21
PC	$\underline{3.08} \pm 0.24$	2.55 ± 0.22	$\underline{2.37} \pm 0.22$	$\underline{2.27} \pm 0.20$	2.24 ± 0.20
DSDS	$8.4\times10^3 \pm 1.8\times10^4$	$7.3\times10^3 \pm 4.2\times10^4$	$5.5\times10^2 \pm 1.9\times10^2$	$10.45 \pm 3.3\times10^2$	$\mathbf{2.21} \pm 0.39$
iDSDS	10.78 ± 1.59	$\mathbf{2.40} \pm 0.21$	$\mathbf{2.34} \pm 0.20$	$\mathbf{2.24} \pm 0.18$	$\underline{2.23} \pm 0.18$

Method	NFE 100	NFE 200	NFE 500	NFE 1000	NFE 2000
ALD	6.74 ± 0.07	6.49 ± 0.16	5.99 ± 0.46	5.46 ± 0.73	4.68 ± 0.86
EM	$\underline{5.22} \pm 0.51$	$\mathbf{4.19} \pm 0.60$	$\mathbf{3.62} \pm 0.54$	$\mathbf{3.45} \pm 0.52$	$\mathbf{3.38} \pm 0.51$
PC	5.71 ± 0.44	4.51 ± 0.62	$\underline{3.89} \pm 0.58$	$\underline{3.69} \pm 0.58$	3.50 ± 0.55
DSDS	$18.54 \pm 1.3\times10^2$	11.06 ± 2.55	8.90 ± 0.33	5.62 ± 0.47	4.89 ± 0.63
iDSDS	$\mathbf{4.95} \pm 0.33$	$\underline{4.34} \pm 0.52$	4.19 ± 0.56	4.05 ± 0.50	$\underline{3.43} \pm 0.41$



(a) DSDS.



(b) iDSDS.



(c) ALD.



(d) EM.



(e) PC.

Figure 12: Overview of generated samples for MNIST (left) and CIFAR-10 (right) samples with the respective best NFE with regards to FID.