Diffusion Path Samplers via Sequential Monte Carlo

James Matthew Young[1] **Paula Cordero-Encinar**[1]
Sebastian Reich[2] **Andrew B. Duncan**[1] **O. Deniz Akyildiz**[1]
[1]Department of Mathematics, Imperial College London
[2]Institut für Mathematik, Universität Potsdam
{matthew.young20, paula.cordero-encinar22}@imperial.ac.uk,
sereich@uni-potsdam.de, {a.duncan, deniz.akyildiz}@imperial.ac.uk

Abstract

We develop diffusion-based samplers for target distributions known up to a normalising constant. To this end, we rely on the well-known diffusion path that smoothly interpolates between a simple base distribution and the target, popularised by diffusion models. We tackle the score estimation problem by developing an efficient sequential Monte Carlo sampler that evolves auxiliary variables from conditional distributions along the path, providing principled score and density estimates for time-varying distributions. To control the variance of score estimates, we further propose practical control variate schedules that incur minimal overhead. We adapt this general framework to paths induced by the Ornstein–Uhlenbeck (OU) time-reversal process, stochastic interpolants, and diffusion annealed Langevin dynamics, outlining their trade-offs. Finally, we provide theoretical guarantees and empirically demonstrate the effectiveness of our method on several synthetic and real-world datasets. Our code is available at: https://github.com/akyildiz-group/dpsmc.

1 Introduction

Sampling from a target distribution $\pi \propto \exp(-V_\pi)$ known up to a normalising constant is a fundamental problem in many fields of science. It underpins Bayesian inference [26], machine learning [7], statistical physics [42], finance [28], computational biology [57], and many others. Despite this, sampling directly from complex targets remains a notoriously difficult task. A standard technique is to introduce an auxiliary path of distributions that bridges a simple base distribution and the target [39, 22]. This allows the sampler to gradually adapt to the complexity of the target distribution. A popular idea in this line of work is to consider the *geometric annealing path* [25, 39]. While convenient, such paths can distort relative mode weights, leading to torpid mixing or *mass teleportation* symptoms [58, 51, 17, 38, 44], and degrade log-Sobolev and related functional inequality constants [10, 15]. These issues collectively hinder the efficiency of sampling algorithms along such paths.

In this work, we focus on a powerful alternative, the *diffusion path*, which has recently gained significant attention in generative modelling [48, 33, 49]. As opposed to the geometric path, the diffusion path is well behaved: under mild assumptions, it admits uniformly bounded functional inequalities and finite action [15]. Their mathematical regularity, combined with the celebrated performance of diffusion models, make the diffusion path an appealing choice for sampling and the centre of a number of recent contributions (see Section 2.1). We expand this area by developing a novel, efficient, and highly-parallelisable sampler on diffusion paths. Specifically:

(C1) We develop a framework for score estimation, building on sequential Monte Carlo (SMC) samplers [22] that sample auxiliary variables sequentially, leading to highly parallelisable samplers.

(C2) We introduce novel control variate schedules for minimising the variance of score estimates, describing their optimum and providing practical estimates that take on minimal overhead.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

(C3) We provide a new bound on the score estimation error, extending the results of He et al. [32], and establish convergence guarantees for our algorithms.

(C4) We demonstrate the effectiveness of our sampler applied to different diffusion dynamics across synthetic and real-world benchmarks, including the task of estimating normalising constants.

Notation. We denote by $\|\cdot\|$ the Euclidean norm for vectors and $\mathrm{B}(m, R)$ a ball centred at m with radius R. We write by $\mathcal{N}(x; m, \mathbf{C})$ the density of a Gaussian random variable with mean m and covariance matrix $\mathbf{C}$ evaluated at x. We let W_2 denote the Wasserstein-2 distance between probability measures [55]: $W_2^2(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu,\nu)} \int \|x - y\|^2 \mathrm{d}\gamma(x, y)$, where $\Gamma(\mu, \nu)$ is the set of couplings between μ and ν. We denote by $\mathrm{KL}(\mu \parallel \nu)$ the Kullback-Leibler (KL) divergence between probability measures μ and ν. Finally, we write the total variance as $\mathrm{Var}_p[f(X)] := \mathrm{Tr}(\mathrm{Cov}_p[f(X)])$.

2 Technical Background

Recent advances in diffusion models [48, 33, 49] have demonstrated that paths based on diffusion processes scale well in high-dimensional and multimodal settings. These paths, which we refer to as *diffusion paths*, can be defined as a continuous-time family of distributions $(\mu_t)_{t\in[0,1]}$ interpolating between a base and target distribution [9, 10, 15], i.e. $\mu_0 = \nu$ and $\mu_1 = \pi$, written as:

$$\mu_t(x) = \tfrac{1}{\sqrt{1-\lambda_t}^d} \nu\Big(\tfrac{x}{\sqrt{1-\lambda_t}}\Big) * \tfrac{1}{\sqrt{\lambda_t}^d} \pi\Big(\tfrac{x}{\sqrt{\lambda_t}}\Big), \tag{1}$$

for $t \in [0, 1]$ and a monotonically increasing schedule $\lambda_t \in [0, 1]$ with $\lambda_1 = 1$. This marginal coincides with the law of $I_t = \sqrt{1-\lambda_t} Z + \sqrt{\lambda_t} X$, where $X \sim \pi$ and $Z \sim \nu$ independent—a spatially linear stochastic interpolant [3]. Throughout, we will assume a Gaussian base distribution $\nu = \mathcal{N}(0, \sigma^2 \mathbf{I})$, for some $\sigma > 0$, which further makes the connection with diffusion models clear.

We may instantiate the path in (1) with two stochastic differential equations (SDEs): (i) a diffusion process that matches the path marginals exactly, and (ii) diffusion-annealed Langevin dynamics (DALD) [15], which approximates the path with an error that can be controlled by a parameter ϵ.

Diffusion path SDE. The diffusion path marginals in (1) are precisely induced by the SDE

$$\mathrm{d}X_t = \Big[\tfrac{\dot\lambda_t}{2\lambda_t} X_t + \Big(\sigma^2 \tfrac{\dot\lambda_t}{2\lambda_t} + \gamma_t\Big) \nabla \log \mu_t(X_t)\Big] \mathrm{d}t + \sqrt{2\gamma_t} \mathrm{d}W_t, \tag{2}$$

for $t \in [0, 1]$, $\gamma_t \geq 0$, and $X_0 \sim \mu_0$ (see Appendix D.3). The time-reversal of the Ornstein–Uhlenbeck (OU) process can then be recovered by choosing $\gamma_t = \sigma^2 \dot\lambda_t / (2\lambda_t)$ and $\lambda_t = \exp(-2T(1-t)) > 0$, where $T > 0$, yielding $\dot\lambda_t / \lambda_t = 2T$. However, the OU process requires an infinite time horizon to reach its invariant Gaussian measure, and consequently T has to be chosen large so $\mu_0 \approx \nu$. In fact, the SDE in (2) also accommodates schedules that reach both endpoints in finite time [3], i.e. $\lambda_0 = 0$. The drift in (2) appears singular at $t = 0$ unlike OU-type exponential schedules with constant $\dot\lambda_t / \lambda_t$, but it can be alleviated by breaking down the score and choosing a suitable schedule as we show later.

Diffusion-annealed Langevin dynamics. The SDE in (2) adopts a convenient form due to the Gaussian base distribution. More generally, for arbitrary base distributions, we can consider diffusion-annealed Langevin dynamics (DALD) [15] which runs the SDE

$$\mathrm{d}X_t = \tfrac{1}{\epsilon} \nabla \log \mu_t(X_t) \mathrm{d}t + \sqrt{\tfrac{2}{\epsilon}} \mathrm{d}W_t, \tag{3}$$

where $t \in [0, 1]$, $X_0 \sim \nu$, and $\epsilon \in (0, 1]$. Here, ϵ controls how closely the SDE's true marginals track the reference path $(\mu_t)_{t\in[0,1]}$, i.e. a smaller ϵ results in dynamics that remain closer to the instantaneous target distributions. Both (2) and (3) have practical trade-offs which we discuss later.

Score formulae. Central to the implementation of these processes is access to score functions $(\nabla \log \mu_t)_{t\in[0,1]}$. The path's convolutional structure admits two expressions for the score, known as the denoising score identity (DSI) [56] and target score identity (TSI) [18], respectively

$$\nabla \log \mu_t(x) = \mathbb{E}_{\varrho_{t,x}} \left[\tfrac{\sqrt{\lambda_t} Y - x}{\sigma^2(1-\lambda_t)}\right] = \mathbb{E}_{\varrho_{t,x}} \left[\tfrac{-\nabla V_\pi(Y)}{\sqrt{\lambda_t}}\right], \tag{4}$$

where $V_\pi = -\log \pi$ is the target potential and $\varrho_{t,x}$ is the so-called posterior (conditional) distribution of the auxiliary (clean) variable Y given (the noisy value) x:

$$\varrho_{t,x}(y) \propto \nu\left(\tfrac{x - \sqrt{\lambda_t} y}{\sqrt{1-\lambda_t}}\right) \pi(y). \tag{5}$$

While the score is generally intractable, it can be learned by a neural network [1, 53, 45, 13, 60, 43] or estimated with Monte Carlo [34, 14, 29, 47, 32, 16]. We follow the latter approach in this work and provide a discussion on related neural samplers in Appendix A.1.

2.1 Related Work

Diffusion-based samplers using Monte Carlo. Huang et al. [34] follow the time-reversed OU process and use importance sampling and Langevin MCMC to estimate the score. They initialise samples using a Langevin-within-Langevin (LL) procedure by running unadjusted Langevin algorithm (ULA) to target μ_0 using Monte Carlo score estimates which are themselves obtained via ULA. Grenioux et al. [29] builds upon this work by introducing a stochastic localisation framework with flexible denoising schedules and a discretisation scheme based on the signal-to-noise ratio (SNR) of the observation process. They start sampling at a later time, a hyperparameter, when both the path marginal and posterior are approximately log-concave. The reliance of these methods on MCMC, however, necessitates equilibration at every time step, creating a sequential bottleneck that cannot be overcome by simply increasing sample size. In an alternative line of work, He et al. [32] propose rejection sampling for score estimation, only making zero-order queries to the target density. While effective in low dimensions, the complexity of rejection sampling grows exponentially with the dimension, even for log-concave targets. Notably, the methods discussed all incur errors coming from their score estimation or SDE discretisation. Introducing a wrapper around reverse-diffusion based samplers, Wu et al. [59] use the auxiliary variables from the posterior to estimate the path marginal density. From which, they propose a nested SMC sampler, RDSMC, where samples can be resampled to better adhere to the diffusion path marginals. In contrast, we employ an SMC sampler *only* for the auxiliary samples. While fundamentally different, our scheme provides unbiased estimates for the path marginal density, allowing it to be nested inside RDSMC.

Score identities. Some works have found convex combinations of the score identities to be beneficial towards the variance of loss objectives. The mixed score identity (MSI) [18, 44, 31] is one example, assigning weights $1 - \lambda_t$ and λ_t to DSI and TSI, respectively. This choice is intuitive as the identities become unstable at opposite ends. A piecewise schedule is similarly justified [16]. This leads us to examine more general schedules, namely those minimising the variance of score estimates. Concurrent to our work, Kahouli et al. [35] and Ko and Geffner [37] propose a similar control variate score identity in the reverse diffusion context, opting for position dependent schedules [35] or learning the schedule itself [37]. We minimise the variance in expectation and consider matrix coefficients.

3 Diffusion Path Sequential Monte Carlo

3.1 Discretisation of Diffusion Path Dynamics

We describe here two schemes to numerically simulate the diffusion path dynamics in (2) and (3).

Initialisation. As the path may start at some $\lambda_0 > 0$, we consider a Langevin-within-Langevin (LL) initialisation [29]. We discretise the time interval as $0 = t_0 = \cdots = t_{N_{\mathrm{LL}}} < \ldots < t_K = 1$, freezing the time index for N_{LL} steps, akin to doing LL. The remaining time steps are discretised uniformly, i.e. $t_k = (k - N_{\mathrm{LL}})/(K - N_{\mathrm{LL}})$, leading to a sequence of distributions $(\mu_{t_k})_{k=0}^K$. To start, samples are drawn from the base distribution and undergo ULA updates with step size h_{LL} for $k < N_{\mathrm{LL}}$.

For $k \geq N_{\mathrm{LL}}$, we discretise the SDEs in (2) or (3) with step size $h = 1/(K - N_{\mathrm{LL}})$ as follows.

Discretisation of diffusion path SDE (2)**.** We use the OU schedule and the exponential integrator (EI) [23], yielding the update rule

$$X_{k+1} = e^{hT} X_k + 2\sigma^2 (e^{hT} - 1) \nabla \log \mu_{t_k}(X_k) + \sigma \sqrt{e^{2hT} - 1} \xi_k, \tag{6}$$

where $\xi_k \sim \mathcal{N}(0, \mathbf{I})$. We also consider a different schedule but postpone our discussion to Section 3.5.

Discretisation of DALD (3)**.** We use an Euler-Maruyama (EM) discretisation, resulting in the update

$$X_{k+1} = X_k + \tfrac{h}{\epsilon} \nabla \log \mu_{t_k}(X_k) + \sqrt{\tfrac{2h}{\epsilon}} \xi_k. \tag{7}$$

We denote these updates by a generic Markov kernel $\mathsf{K}_{k+1}^{\mathrm{sde}}$. Note the score function can be written as

$$\nabla \log \mu_{t_k}(x) = \mathbb{E}_{\varrho_{t_k,x}}[\varphi_{k,x}(Y)], \tag{8}$$

for some vector-valued test function $\varphi_{k,x} : \mathbb{R}^d \to \mathbb{R}^d$ (choices of which can be judiciously made as we discuss in Section 3.4) and $\varrho_{t_k,x}$, the posterior distribution defined in (5). The time-varying expectation in (8) naturally suggests an SMC scheme for score estimation, which we describe next.

3.2 SMC Samplers for Score Estimation

We define a sequence of target distributions $(\rho_k)_{k=0}^K$, given by $\rho_k := \varrho_{t_k,X_k}$, where X_k is our sample at time t_k following our discretised dynamics in (6) or (7). We then define the unnormalised target and proposal distributions on the (extended) path space $\mathbb{R}^{d(k+1)}$

$$\tilde{\rho}_{0:k}(y_{0:k}) := \tilde{\rho}_k(y_k)\prod_{p=1}^k \mathsf{L}_{p-1}(y_p, y_{p-1}),$$
$$q_{0:k}(y_{0:k}) := q_0(y_0)\prod_{p=1}^k \mathsf{K}_p(y_{p-1}, y_p),$$

where $(\mathsf{K}_p)_{p=1}^k$ and $(\mathsf{L}_p)_{p=0}^{k-1}$ are a sequence of forward and backward Markov kernels,

$$\tilde{\rho}_k(y) = \frac{1}{\sqrt{1-\lambda_{t_k}}^d}\nu\left(\frac{X_k - \sqrt{\lambda_{t_k}}y}{\sqrt{1-\lambda_{t_k}}}\right)\pi(y),$$

and q_0 is some initial auxiliary distribution. It can be readily checked that the y_k-marginal of (normalised) $\rho_{0:k}$ is indeed ρ_k, justifying the choice of the extended target. The resulting importance weights can be written in a recursive fashion

$$\begin{aligned} W_k(y_{0:k}) &:= \frac{\tilde{\rho}_{0:k}(y_{0:k})}{q_{0:k}(y_{0:k})} \\ &= W_{k-1}(y_{0:k-1})G_k(y_{k-1}, y_k), \end{aligned}$$

with the potential $G_k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$:

$$G_k(y_{k-1}, y_k) = \frac{\tilde{\rho}_k(y_k)\mathsf{L}_{k-1}(y_k, y_{k-1})}{\tilde{\rho}_{k-1}(y_{k-1})\mathsf{K}_k(y_{k-1}, y_k)}.$$

We then have the following result through the Feynman-Kac formulae [19, 20].

Algorithm 1 DPSMC (Generic)

input base variance σ^2, initial aux. distribution q_0, particle count N, disc. step count K, LL step count N_{LL}, LL step size h_{LL}, test function $\varphi_{k,x}$, ESS threshold κ, and SDE forward kernel $\mathsf{K}_k^{\mathrm{sde}}$

Initialisation

$X_0 \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, $Y_0^i \sim q_0$ for all $i \in [N]$
$W_0^i = \tilde{\rho}_0(Y_0^i)/q_0(Y_0^i)$ for all $i \in [N]$
$w_0^i = W_0^i / \sum_{j=1}^N W_0^j$ for all $i \in [N]$
$S_0^N = \sum_{i=1}^N w_0^i \varphi_{0,X_0}(Y_0^i)$

Main Loop

for $k = 1, \ldots, K-1$ **do**
  if $k < N_{\mathrm{LL}}$ **then**
    $X_k \sim \mathcal{N}(X_{k-1} + h_{\mathrm{LL}} S_{k-1}^N, 2h_{\mathrm{LL}}\mathbf{I})$ {LL}
  else $X_k \sim \mathsf{K}_k^{\mathrm{sde}}(\cdot \mid X_{k-1}, S_{k-1}^N)$ {SDE step}
  for $i = 1, \ldots, N$ **do**
    $\bar{Y}_k^i \sim \mathsf{K}_k(\cdot \mid Y_{k-1}^i, X_k)$
    $W_k^i = W_{k-1}^i \frac{\tilde{\rho}_k(\bar{Y}_k^i)\mathsf{L}_{k-1}(\bar{Y}_k^i, Y_{k-1}^i)}{\tilde{\rho}_{k-1}(Y_{k-1}^i)\mathsf{K}_k(Y_{k-1}^i, \bar{Y}_k^i)}$
  end for
  $w_k^i = W_k^i / \sum_{j=1}^N W_k^j$ for all $i \in [N]$
  $S_k^N = \sum_{i=1}^N w_k^i \varphi_{k,X_k}(\bar{Y}_k^i)$ {Score estimation}
  if $\widehat{\mathrm{ESS}}(\{w_k^i\}_{i=1}^N) < \kappa N$ **then**
    $\{Y_k^i\}_{i=1}^N \sim \sum_{j=1}^N w_k^j \delta_{\bar{Y}_k^j}$ {Resampling}
    $W_k^i = 1$ for all $i \in [N]$
  else $Y_k^i = \bar{Y}_k^i$ for all $i \in [N]$
end for
$X_K \sim \mathsf{K}_K^{\mathrm{sde}}(\cdot \mid X_{K-1}, S_{K-1}^N)$ {SDE step}

output approximate sample from the target X_K

Proposition 1. *For any $k \geq 1$, let $Y_{0:k} \sim q_{0:k}$. If $\varphi_{k,x}$ forms a score identity as in* (8)*, then we have*

$$\nabla \log \mu_{t_k}(X_k) = \frac{\mathbb{E}[\varphi_{k,X_k}(Y_k)\prod_{p=0}^k G_p(Y_{p-1}, Y_p)]}{\mathbb{E}[\prod_{p=0}^k G_p(Y_{p-1}, Y_p)]}, \quad \mu_{t_k}(X_k) = \mathbb{E}\left[\prod_{p=0}^k G_p(Y_{p-1}, Y_p)\right]$$

where $G_0(y_{-1}, y_0) := \tilde{\rho}_0(y_0)/q_0(y_0)$ and the expectations are taken with respect to the path measure induced by the forward kernels $(\mathsf{K}_p)_{p=1}^k$ and initial distribution q_0.

See Appendix B.1 for the proof. Proposition 1 therefore expresses the score and marginal density in terms of expectations under the forward path measure, which can be estimated directly with an SMC sampler. Notably, this provides unbiased estimates for the marginal density which makes the scheme compatible with pseudo-marginal or *exact approximation* methods [4, 5, 59] which we demonstrate later. Next, we outline our sampling method based on this construction.

3.3 The DPSMC Sampler

The previous framework leads us to introduce our diffusion path sequential Monte Carlo (DPSMC) sampler, summarised in Algorithm 1. Quite simply, we have a discretised diffusion process whose dynamics are intertwined with an SMC sampler. The state of the process defines the SMC sampler's intermediate target, and the SMC sampler provides a score estimate for updating the process.

We walk through a single iteration of our algorithm to demonstrate. At the end of time t_{k-1}, we have three things: (i) our sample X_{k-1}, a result of integrating the SDE for $k-1$ steps with estimated scores; (ii) a weighted ensemble $\{(W_{k-1}^i, Y_{k-1}^i)\}_{i=1}^N$, an output of our SMC sampler targeting ρ_{k-1}; and (iii) a score estimate S_{k-1}^N, formed by a Monte Carlo approximation of (8) using the weighted ensemble and test function $\varphi_{k-1,X_{k-1}}$. At time t_k, we first update our sample to X_k with an integration step involving the current score estimate. Then, we run one iteration of the SMC sampler to target ρ_k. This involves propagating each Y_{k-1}^i by a forward kernel $\mathsf{K}_k(\cdot \mid Y_{k-1}^i, X_k)$, updating their weights, and resampling, resulting in a new weighted ensemble $\{(W_k^i, Y_k^i)\}_{i=1}^N$ targeting ρ_k. Finally, an updated score estimate S_k^N is produced using the new ensemble and test function φ_{k,X_k}. The procedure is repeated until time t_{K-1}, after which we perform one final integration step to obtain X_K. Note that we perform an LL initialisation for the first N_{LL} steps if $\lambda_0 > 0$, whereby the only change is the underlying diffusion dynamics being discretised is replaced with an overdamped Langevin diffusion.

In the next sections, we carefully investigate design choices to make our generic scheme practical.

3.4 Score Identities

We now look at stable choices for test functions $\varphi_{k,x}$ that form score identities as in (8). Building upon DSI and TSI, as in (4), we introduce a class of test functions

$$\phi_{t,x}^{\mathbf{A}}(y) := \tfrac{1}{\sqrt{1-\lambda_t}}\mathbf{A}\nabla \log \nu\left(\tfrac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right) + \tfrac{1}{\sqrt{\lambda_t}}(\mathbf{I}-\mathbf{A})\nabla \log \pi(y), \tag{9}$$

for $t \in [0,1]$ and $\mathbf{A} \in \mathbb{R}^{d\times d}$, which indeed satisfy $\nabla \log \mu_t(x) = \mathbb{E}_{\varrho_{t,x}}[\phi_{t,x}^{\mathbf{A}}(Y)]$ (see Proposition A.1 for a proof). In practice, we set $\varphi_{k,x} = \phi_{t_k,x}^{\mathbf{A}}$ at time step k. We are interested in the variance behaviour of these test functions under the true marginals, parameterised by matrix schedules $\mathbf{A}_t \in \mathbb{R}^{d\times d}$. We refer to these as *control variate* (CV) schedules, given we can arrive at the same test function in (9) by introducing a control variate matrix (see Remark A.1). We first consider the case when $\mathbf{A}_t = \alpha_t \mathbf{I}$, i.e. a scalar weighting of DSI and TSI. The following result characterises the optimal choice of α_t for minimising the *expected* variance of the corresponding score estimator.

Proposition 2. *The scalar CV schedule $\alpha_t^* \in \mathbb{R}$ minimising the expected variance is given by*

$$\alpha_t^* = \operatorname*{argmin}_{\alpha\in\mathbb{R}} \mathbb{E}_{X\sim\mu_t}[\mathrm{Var}_{Y\sim\varrho_{t,X}}[\phi_{t,X}^{\alpha\mathbf{I}}(Y)]] = \frac{\frac{1}{\lambda_t}\mathrm{Var}_\pi[\nabla \log \pi(X)]}{\frac{1}{1-\lambda_t}\mathrm{Var}_\nu[\nabla \log \nu(X)] + \frac{1}{\lambda_t}\mathrm{Var}_\pi[\nabla \log \pi(X)]}.$$

See Appendix C.2.1 for the proof. This expression decouples the dependence of time from statistics of the endpoint distributions. It aligns with our intuition of avoiding the instabilities at both ends, having α_t^* be non-increasing and going from one to zero. Unlike heuristic choices that start with DSI and end with TSI, this result shows explicitly when one is favoured, e.g. it is possible that $\alpha_t^* \approx 1$ for majority of the time when the target has much larger score variance than the base. When they coincide in score variance, we recover $\alpha_t^* = 1 - \lambda_t$, i.e. MSI. For matrices, we have the next result.

Proposition 3. *The matrix CV schedule $\mathbf{A}_t^* \in \mathbb{R}^{d\times d}$ minimising the expected variance is given by*

$$\mathbf{A}_t^* = \operatorname*{argmin}_{\mathbf{A}\in\mathbb{R}^{d\times d}} \mathbb{E}_{X\sim\mu_t}[\mathrm{Var}_{Y\sim\varrho_{t,X}}[\phi_{t,X}^{\mathbf{A}}(Y)]] = \tfrac{1}{\lambda_t}\mathcal{I}_\pi\left(\tfrac{1}{1-\lambda_t}\mathcal{I}_\nu + \tfrac{1}{\lambda_t}\mathcal{I}_\pi\right)^{-1},$$

where $\mathcal{I}_p := \mathrm{Cov}_p[\nabla \log p(X)] = \mathbb{E}_p[\nabla \log p(X) \nabla \log p(X)^\top]$ denotes the score covariance of p.

See Appendix C.2.3 for the proof. This optimal weighting accounts for the pairwise covariances of individual score components. Note the scalar schedule averages them out. We therefore expect a discrepancy between the scalar and matrix schedules when the base and target have a covariance mismatch, e.g. for extremely anisotropic targets. An example is given in Figure 1, documenting the MSE evolution of different score identities. Ideally, one would choose base densities with a similar covariance structure as the target to maximise the effectiveness of scalar schedules.

In practice, we need estimates of varying degrees for the target score covariance, i.e. either its trace, diagonal, or the full matrix. By Lemma A.4 in Appendix, we can write it as

$$\mathcal{I}_\pi = \mathbb{E}_{X\sim\mu_t}[\mathbb{E}_{Y\sim\varrho_{t,X}}[\nabla \log \pi(Y) \nabla \log \varrho_{t,X}(Y)^\top]],$$

which is amenable to Monte Carlo estimation. We use the empirical distribution formed by our samples as a proxy for μ_t and our auxiliary variables for the inner expectation. This incurs no additional energy evaluations, as they can be reused for the score estimate.

Following Kahouli et al. [35], we call the identities formed by our schedules as *marginalised control-variate score identities* (MCVSI) and refer to the matrix case as MMCVSI. Note that we can minimise the estimator's variance for each sample x but do so in expectation. This admits an interpretable form, indicating trade-offs between scalar, diagonal, and full-matrix control-variate coefficients, while practically being more stable, as the target score covariance is a static unknown and all auxiliary variables can be used for estimation. Further, while our score estimates via SMC are biased due to self-normalisation, we empirically find our schedules consistently improve the score estimates. This can be justified, since the variance of the unbiased estimator coincides with the asymptotic variance of the self-normalised estimator when the proposal matches the target. See Appendix H.1 for ablations on score identities.

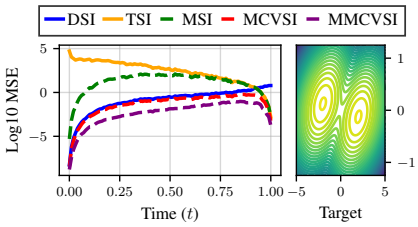

Figure 1: *Average MSE of different score identities using exact samples from diffusion path marginals and auxiliary variables from their conditional distribution.* MCVSI and MMCVSI consistently remain under the DSI/TSI lower envelope.

3.5 Diffusion Path Design

Our diffusion path is parameterised by the base Gaussian variance σ^2 and the schedule λ_t. It is not surprising that poor choices for these parameters can render the sampler inefficient.

Base Gaussian variance. If σ^2 is too small, then $\mu_t(x) \approx \pi\left(x/\sqrt{\lambda_t}\right)/\sqrt{\lambda_t}^d$, causing the mode structure to be determined within a few time steps and the remainder of the path spent dilating the modes. If it is too large, the path marginals become indistinguishable from a Gaussian for most of the path. To get around this, we consult the SNR of our stochastic process. We define

$$\mathrm{SNR}(t) := \frac{\mathbb{E}[\|\sqrt{\lambda_t}X\|^2]}{\mathbb{E}[\|\sqrt{1-\lambda_t}Z\|^2]} = \frac{\mathbb{E}_\pi[\|X\|^2]\lambda_t}{\sigma^2 d(1-\lambda_t)},$$

using the power or second moment of the target $\mathbb{E}_\pi[\|X\|^2]$ as opposed to its scalar variance [29]. Here, choosing $\sigma^2 \propto \mathbb{E}_\pi[\|X\|^2]/d$ yields an SNR independent of the target. We set $\sigma^2 = \mathbb{E}_\pi[\|X\|^2]/d$, as it is the most natural choice, keeping the second moment fixed across the path marginals.

Diffusion path schedule. We seek a λ_t schedule for DALD and an alternative to the OU schedule for the exact dynamics in (2). Intuitively, the DALD schedule should spend more time where the path is rapidly changing and less where it does not. Let $\mu = (\mu_t)_{t\in[0,1]}$ denote the curve of probability measures corresponding to the diffusion path. We can consider the (Wasserstein) action functional $\mathcal{A}$ as a measure of how rapidly the path marginals evolve throughout the curve. This quantity upper bounds the KL divergence between the target and final marginal distribution of DALD [30, 15] (see Appendix D.1 for details). While choosing λ_t to minimise it is a natural choice, the action is generally intractable, so we instead minimise the upper bound functional

$$\mathcal{A}(\mu) := \int_0^T |\dot{\mu}|_t^2 \mathrm{d}t \leq \int_0^T \left[\frac{\sigma^2 d}{4(1-\lambda_t)} + \frac{\mathbb{E}_\pi[\|X\|^2]}{4\lambda_t}\right]\dot{\lambda}_t^2 \mathrm{d}t, \tag{10}$$

where $|\dot{\mu}|_t = \lim_{\delta\to 0} W_2(\mu_{t+\delta}, \mu_t)/|\delta|$ is the metric derivative of μ at time t. The minimiser is available analytically when $\sigma^2 = \mathbb{E}_\pi[\|X\|^2]/d$ and is given by

$$\lambda_t^* = \sin^2\left(\tfrac{\pi t}{2}\right) = 1 - \cos^2\left(\tfrac{\pi t}{2}\right),$$

which is the well-known cosine schedule (see Appendix D.2 for the proof). This schedule eases in and out, allowing more time at the start for samples to localise in modes and at the end for details to be gradually refined. This also provides a handy finite upper bound on the action.

We previously hinted that a non-exponential schedule with $\lambda_0 = 0$ can be chosen to admit the exact SDE in (2), despite the seemingly singular drift coefficient. With DSI, the drift can be rewritten as

$$b_t(X_t) := \tfrac{\dot{\lambda}_t}{2\lambda_t}[X_t + \sigma^2\nabla \log \mu_t(X_t)] = \tfrac{\dot{\lambda}_t}{2\sqrt{\lambda_t}}\mathbb{E}_{\varrho_{t,X_t}}\left[\tfrac{Y-\sqrt{\lambda_t}X_t}{1-\lambda_t}\right],$$

and now we simply require $\dot{\lambda}_t/\sqrt{\lambda_t}$ remaining finite at $t = 0$. While there are several options, we set $\gamma_t = \sigma^2$ and $\lambda_t = \sin^2(\frac{\pi t}{2})$ to match the diffusion coefficient of the OU process while following the curve at the same speed as DALD. This way, we can more closely examine how the dynamics differ under similar conditions. Notably, this allows us to start exactly from a Gaussian while preserving exact dynamics, rendering it free from the critical hyperparameters T and ϵ that alter the diffusion process. We refer to this as following a stochastic interpolant (SI) dynamics [3].

3.6 Algorithmic Choices

We briefly detail our implementation of the DPSMC algorithm in practice.

Initialisation. A seemingly pathological quality shared by diffusion-based samplers is the fact that $\varrho_{0,x} = \pi$ (see Figure 2). To see this, one can think of $\varrho_{t,x}$ as the distribution of where a (noisy) sample x at time t will end up at time $t = 1$. This also implies $\varrho_{1,x} = \delta_x$. Notably, under some conditions, there exists a time window where both path marginals and posteriors are log-concave or approximately so [29]. Hence, starting at a later time not at a Gaussian, e.g. in this window, may be more of a practical choice rather than a disadvantage. As such, we search for λ_0 in $[0, 1]$ uniformly and perform an LL initialisation for OU dynamics. Our other schemes with $\lambda_0 = 0$ sidestep the need for LL but require sampling from $\rho_{0,x}$. Indeed, in the next section, we show that initial posterior samples can be rather poor and still produce good score estimates. We choose $q_0 = \mathcal{N}(0, \tilde{\sigma}^2 \min(1, (1-\lambda_0)/\lambda_0)\mathbf{I})$ to initialise the auxiliary variables. We follow Grenioux et al. [29] in setting $\tilde{\sigma}^2 = R^2 d + \tau^2$, where we assume $\pi = \mathcal{N}(0, \tau^2\mathbf{I}) * \eta$ for η compactly supported on $\mathrm{B}(\mathbb{E}_\pi[X], R^2 d)$. We remark that LL for $\lambda_0 > 0$ incurs an initialisation bias that appears later in the overall algorithm error.

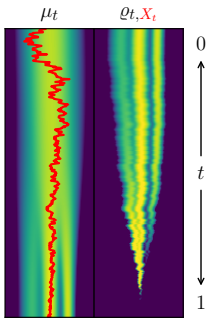

Figure 2: *Evolution of path marginal and posterior.* In red is a sample trajectory.

Propagation. We propagate auxiliary variables according to a Metropolis-adjusted Langevin algorithm (MALA) kernel $\mathsf{K}_k^{\mathrm{MALA}}$ with an adaptive step size, adjusted to target an acceptance ratio. We make the standard choice of backward kernels being reversed kernels in relation to the forward kernels [22, Section 3.3.2.3], i.e. $\mathsf{L}_{k-1}(y_k, y_{k-1}) = \rho_k(y_{k-1})\mathsf{K}_k(y_{k-1}, y_k)/\rho_k(y_k)$, simplifying the weight update (see Appendix B.2). We also use stratified resampling, a low-variance resampling method, and trigger it adaptively with threshold $\kappa = 0.5$. With our samples and auxiliary variables, we estimate the matrix schedule in Proposition 3 and hence the scores with the test function in (9).

Our practical implementation of DPSMC is given in Algorithm 2 in Appendix. While our samples are interacting in step size adaptation and target score covariance estimation, one could first estimate these underlying quantities and consider the non-interacting version of our sampler with these settings.

4 Theoretical Analysis

In this section, we analyse the different sources of error in our algorithm, building on the theoretical analysis presented in [6, 15]. We first introduce the required assumptions.

Assumption 1. *The target distribution has density w.r.t. Lebesgue, $\pi \propto \exp(-V_\pi) \in \mathcal{P}(\mathbb{R}^d)$, has a finite second-order moment, $M_2 := \mathbb{E}_\pi[\|X\|^2] < \infty$, and the potential V_π is L_π-smooth.*

Assumption 2. *The potential V_π is strongly convex outside a compact set, i.e., $\exists\, R > 0$ such that $\inf_{\|x\| \geq R} \nabla^2 V_\pi \succ 0$.*

We note that under Assumptions 1 and 2, it follows from Cordero-Encinar et al. [15, Lemma 3.2] and Cattiaux et al. [8, Lemma 3.8] that $\nabla \log \mu_t$ is Lipschitz continuous with finite constant L_t, satisfying $\sup_t L_t \leq L$. A key ingredient to bound the KL divergence between the target π and the final distribution p_1 obtained by the DPSMC algorithm is to control the score estimation error. First, assume there is no discretisation error, so that we can sample $X_k \sim \mu_{t_k}$ exactly and then generate the weighted particles $\{(w_k^i, \bar{Y}_k^i)\}_{i=1}^N$. For simplicity, let α_t be an arbitrary scalar schedule and define the score estimator

$$S_k^N(X_k) := \sum_{i=1}^N w_k^i \phi_{t_k, X_k}^{\alpha_{t_k}\mathbf{I}}(\bar{Y}_k^i). \tag{11}$$

The following proposition provides a bound on the error of this estimator.

Proposition 4 (Score estimation error)**.** *Suppose Assumption 1 holds and $0 < \lambda_{t_k} < 1$. Let $\{X_k\}_{k=1}^K$ be such that $\mu_{t_k} = \mathrm{Law}(X_k)$ for all $k \geq 1$. Given the score estimator $S_k^N(X_k)$ defined in (11), assume that for some $\delta(k, N) \geq 0$, $\mathbb{E}\left[W_2^2\left(\sum_{i=1}^N w_k^i \delta_{\bar{Y}_k^i}, \varrho_{t_k, X_k}\right)\right] \leq \delta(k, N)$ holds where the expectation is over the joint law of X_k, the particles, and the weights. Then*

$$\mathbb{E}\left[\|\nabla \log \mu_{t_k}(X_k) - S_k^N(X_k)\|^2\right] \leq 2\,\delta(k, N)\left(\alpha_{t_k}^2 \frac{\lambda_{t_k}}{\sigma^4(1-\lambda_{t_k})^2} + (1-\alpha_{t_k})^2 \frac{L_\pi^2}{\lambda_{t_k}}\right). \tag{12}$$

See Appendix F.1 for the proof. This result generalises Proposition 3.1 in [32], which can be recovered by setting $\alpha_{t_k} = 1$ for all k and using a system of equally weighted i.i.d particles. To see how this relates to posterior sampling difficulties at the start, consider the regime when t_k is small. Here, λ_{t_k} is small and $\alpha_{t_k} = \alpha^*_{t_k}$ is close to 1. The bracketed terms are hence of order $\mathcal{O}(\lambda_{t_k})$ which implies that having $\delta(k, N) = \mathcal{O}(\varepsilon^2/\lambda_{t_k})$ provides a bound on the first term of order $\mathcal{O}(\varepsilon^2)$. For example, when $\alpha_{t_k} = 1 - \lambda_{t_k}$, i.e. MSI, the bracketed terms become $\lambda_{t_k}(1/\sigma^4 + L_\pi^2)$. Hence, while ϱ_{t_k, X_k} is hard to sample from for small t_k, relatively large mean-squared approximation errors (of order $\mathcal{O}(\varepsilon^2/\lambda_{t_k})$) between ϱ_{t_k, X_k} and the distribution of our auxiliary variables can still result in a reliable score estimator when measured in the L^2 sense. In other words, poorly sampled SMC particles in the initial steps do not significantly degrade the score error. We remark, however, that it remains non-trivial how N controls the W_2^2 distance between the SMC empirical measure and the posterior.

We can apply Proposition 4 to obtain a bound on the score estimation error for our weighted estimator (11). In particular, if we require the score estimation error over time to be bounded by $\varepsilon^2_{\text{score}}$, then assuming $\delta(k, N)$ vanishes with large N for all k, we can choose N large enough to satisfy

$$\sum_{k=0}^{K-1} h\, \mathbb{E}\left[\left\|\nabla \log \mu_{t_k}(X_k) - S_k^N(X_k)\right\|^2\right] \leq \varepsilon^2_{\text{score}}.$$

This allows us to invoke Benton et al. [6, Theorem 1] and Cordero-Encinar et al. [15, Theorem 3.4] to bound the final errors due to initialisation, discretisations, or imperfect tracking. In particular, we have the following result.

Theorem 1. *Under Assumption 1 and further Assumption 2 for (ii) and (iii), the DPSMC algorithm with (i) OU time-reversal dynamics initialised at $X_{N_{LL}} \sim p_0$, (ii) SI dynamics initialised at $X_0 \sim \nu$, and (iii) DALD initialised at $X_0 \sim \nu$, satisfy respectively*

$$\begin{aligned}
&(i)\ \mathrm{KL}\left(\pi \parallel p_1\right) \lesssim \mathrm{KL}(\mu_0 \parallel p_0) + \tfrac{d}{K - N_{LL}}\left(1 + T\right) + \varepsilon^2_{score},\\
&(ii)\ \mathrm{KL}\left(\pi \parallel p_1\right) \lesssim \tfrac{dL^2}{K} + \varepsilon^2_{score},\\
&(iii)\ \mathrm{KL}\left(\pi \parallel p_1\right) \lesssim \epsilon\left(1 + \tfrac{L^2}{K^2\epsilon^4}\right)(M_2 + d) + \tfrac{dL^2}{K\epsilon^2}\left(1 + \tfrac{L}{K\epsilon}\right) + \varepsilon^2_{score}.
\end{aligned}$$

See Appendix F.2 for a proof. These bounds imply that, for $\varepsilon = \mathcal{O}(\varepsilon_{\text{score}})$, under appropriate choices for K, μ_0, N_{LL}, and ϵ, we can keep $\mathrm{KL}\left(\pi \parallel p_1\right) \leq \varepsilon^2$.

5 Experiments

We now evaluate DPSMC across several benchmarks. Tables 1 and 2 summarise our findings, where the mean and standard error of the metrics are reported.

Samplers. We compare our methods against several classical samplers: annealed importance sampling (AIS) [39] and SMC [22], and those based on reverse-diffusion, RDMC [34] and SLIPS [29]. For completeness, we include parallel tempering (PT) [27, 50] as an equilibrium method baseline. Hyperparameter choices are found in Appendix G.1. We fix the number of energy evaluations for all samplers, except RDMC with twice as many. We also globally fix the number of discretisation steps to $K = 1024$. We test several instances of DPSMC, each driven by different dynamics: DALD, DALD with LL, the OU time-reversal, and SI. We set $N_{\text{LL}} = 64$ for methods with LL and $N_Y = 128$ to match the budget. Samplers are assessed on the 4096 samples they produce across 10 seeds.

Toy target distributions. We consider several toy targets that test a sampler's ability to tackle multimodality and complicated geometries. Our first example is GMM40, a uniform mixture of 40 Gaussians with identity covariance and randomly sampled means. We consider dimensions $d = 2$ and $d = 50$. We also test targets with challenging geometries such as Rings ($d = 2$) [29], where regions of high density are localised along four concentric rings, and Funnel ($d = 10$) [40], which consists of an exponentially narrowing funnel. We evaluate samples relative to the target through their W_2 distance for GMM40 and Rings, and the sliced Kolmogorov-Smirnov (KS) distance for Funnel. We find DPSMC yields the best performance in all benchmarks. For Funnel, massive gains are attributed to our matrix control-variates that handle anisotropy well. We find starting later with LL benefits DALD, providing a finer discretisation and an easier initial posterior. On the other hand, SI, free of critical hyperparameters, performs comparably to OU.

Table 1: *Performance of algorithms on toy targets and Bayesian logistic regression datasets across 10 seeds.* Best result is in **bold**, and second-best is underlined. The second column indicates the number of batched energy evaluations—a metric for measuring runtimes in the regime where hardware capacity exceeds batch size requirements. DPSMC is overall performant while highly parallelisable.

Algorithm	Batched Evals (↓)	GMM40 ($d = 2$) W_2 (↓)	GMM40 ($d = 50$) W_2 (↓)	Rings ($d = 2$) W_2 (↓)	Funnel ($d = 10$) KS (↓)	Ionosphere ($d = 35$) Log-LL (↑)	Sonar ($d = 61$) Log-LL (↑)
AIS	131K	$2.35_{\pm 0.41}$	$115.13_{\pm 2.98}$	$0.19_{\pm 0.01}$	$0.037_{\pm 0.003}$	$-87.79_{\pm 0.10}$	$-110.94_{\pm 0.10}$
SMC	131K	$5.50_{\pm 0.22}$	$111.81_{\pm 4.84}$	$\mathbf{0.18_{\pm 0.01}}$	$0.035_{\pm 0.003}$	$-87.79_{\pm 0.10}$	$-110.94_{\pm 0.10}$
RDMC [34]	32K	$12.35_{\pm 0.11}$	$79.37_{\pm 0.10}$	$0.29_{\pm 0.02}$	$0.082_{\pm 0.003}$	$-109.97_{\pm 0.23}$	$-129.82_{\pm 0.18}$
SLIPS [29]	32K	$4.07_{\pm 0.40}$	$27.57_{\pm 0.83}$	$\mathbf{0.18_{\pm 0.01}}$	$0.073_{\pm 0.003}$	$-87.88_{\pm 0.10}$	$-110.88_{\pm 0.06}$
DPSMC (DALD)	**1K**	$1.92_{\pm 0.22}$	$29.05_{\pm 1.51}$	$0.20_{\pm 0.02}$	$0.062_{\pm 0.003}$	$\underline{-83.77_{\pm 0.07}}$	$\underline{-108.45_{\pm 0.05}}$
" w/ LL	**1K**	$1.57_{\pm 0.33}$	$\mathbf{26.82_{\pm 1.38}}$	$0.19_{\pm 0.02}$	$0.046_{\pm 0.004}$	$-85.22_{\pm 0.11}$	$-109.34_{\pm 0.08}$
DPSMC (OU)	**1K**	$\mathbf{1.14_{\pm 0.24}}$	$\underline{27.27_{\pm 1.06}}$	$\mathbf{0.18_{\pm 0.02}}$	$\underline{0.028_{\pm 0.002}}$	$\mathbf{-83.73_{\pm 0.16}}$	$\mathbf{-108.23_{\pm 0.07}}$
DPSMC (SI)	**1K**	$\underline{1.23_{\pm 0.30}}$	$28.78_{\pm 1.78}$	$\mathbf{0.18_{\pm 0.03}}$	$\mathbf{0.026_{\pm 0.004}}$	$-87.54_{\pm 0.11}$	$-111.06_{\pm 0.08}$
PT	2M	$1.13_{\pm 0.18}$	$42.44_{\pm 2.13}$	$0.18_{\pm 0.02}$	$0.022_{\pm 0.005}$	$-87.91_{\pm 0.17}$	$-111.04_{\pm 0.08}$

Table 2: *Performance of algorithms on normalising constant estimation and sample quality on Rings and Funnel targets*. Best result is in **bold**, and second-best is underlined. DPSMC provides improved estimates for the normalising constant. AIS, SMC, and RDSMC results are lifted from Wu et al. [59].

Algorithm	Rings ($d = 2$)			Funnel ($d = 10$)	
	Radius TVD (↓)	Angle TVD (↓)	$\log Z_\pi$ Bias (↓)	KS (↓)	$\log Z_\pi$ Bias (↓)
AIS	$\mathbf{0.10_{\pm 0.00}}$	$\underline{0.15_{\pm 0.00}}$	$0.05_{\pm 0.00}$	$\underline{0.07_{\pm 0.00}}$	$\underline{0.28_{\pm 0.01}}$
SMC	$\mathbf{0.10_{\pm 0.00}}$	$\underline{0.15_{\pm 0.00}}$	$0.05_{\pm 0.00}$	$\underline{0.07_{\pm 0.00}}$	$\underline{0.28_{\pm 0.01}}$
RDSMC	$\underline{0.13_{\pm 0.01}}$	$0.21_{\pm 0.01}$	$\underline{0.03_{\pm 0.00}}$	$0.11_{\pm 0.00}$	$\underline{0.28_{\pm 0.10}}$
DPSMC (OU)	$0.31_{\pm 0.01}$	$\mathbf{0.14_{\pm 0.01}}$	$\mathbf{0.02_{\pm 0.01}}$	$\mathbf{0.06_{\pm 0.00}}$	$\mathbf{0.12_{\pm 0.07}}$

Bayesian logistic regression. To showcase how samplers deal with high-dimensional real-world data, we consider posterior sampling from a Bayesian logistic regression model under the Ionosphere ($d = 35$) and Sonar ($d = 61$) datasets. Each dataset is split 70-10-20% into training, validation, and testing sets. Samples, i.e. model weights, are evaluated on their predictive log-likelihood on the test set. For fairness, we tune hyperparameters only using the validation log-likelihood and combine validation and training sets for samplers that do not require tuning. We find DALD and OU perform best. Here, the target is log-concave and so are the path marginals and posteriors. Despite both starting at a Gaussian, we believe DALD is able to re-equilibrate w.r.t. the path marginals at times when scores are more reliable and hence do not propagate score errors the same way as in SI that relies on exact dynamics. OU achieves a similar performance by initialising near the end of the path.

Normalising constant estimation. An important aspect of samplers is their ability to estimate normalising constants of unnormalised target densities. We adapt DPSMC (OU) to use the RDSMC framework [59], an exact approximation method enabling AIS/SMC on the diffusion path itself, and test it on Rings and Funnel under a comparable number of energy evaluations and discretisation, adjusting the number of MALA steps to match. This is possible as our SMC setup produces unbiased estimates for the path marginal densities (see Proposition (1)) unlike RDMC or SLIPS. Here, we initialise auxiliary variables at ρ_0 with AIS and use our SMC framework thereafter. We measure the total variation distance (TVD) of radial and angular components w.r.t. ground truth samples for Rings and measure KS for Funnel. We find DPSMC achieves superior normalising constant estimates while maintaining good samples. When we deviate from RDSMC's settings, we observe that a finer discretisation with fewer MALA steps leads to better sample quality, e.g. lower radius TVD, at the cost of more biased constant estimates. We can pursue this trade-off as we evolve auxiliary variables along conditional distributions instead of running SMC between a prior and ρ_k at every step.

Overall, we find DPSMC has speed-ups of several orders of magnitude compared to other samplers while achieving comparable if not better sample quality. In the modern age of GPUs and parallel computing, inference at scale is primarily achieved by distributing work horizontally and has significantly rewarded methods that can leverage this [54, 11]. We argue that a sampler's *ability to batch energy*

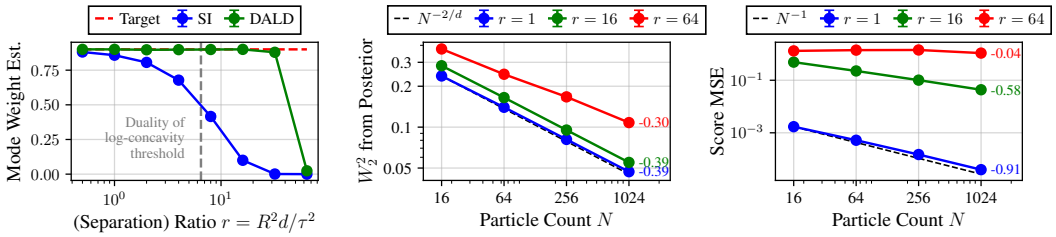

Figure 3: *Ablation experiments on a five-dimensional bi-modal target as mode separation is increased.* (Left) predicted weight of heavier mode when auxiliary variables are initialised at the lighter mode. (Middle and right) empirical rates for how fast the W_2^2 from the posterior and score MSE vanish as particle count is increased. Log-log ordinary least-squares slopes are affixed next to each trend line.

evaluations is hence an increasingly important design axis. In essence, methods that can exploit this batching for estimating better transport paths can achieve better wall-clock times and sample quality. Next, we perform some ablation experiments to test the limitations of our method and assumptions.

Initialisation dependency. Given the challenging posterior at time $t = 0$, we aim to quantify how poor initialisation of auxiliary variables affects our algorithm's performance. We consider sampling from an uneven bi-modal Gaussian mixture target in five dimensions while adversarially initialising auxiliary variables at the lighter mode. Further details and results are in Appendix H.4. We test our SI and DALD instantiations of DPSMC, which start at $\lambda_0 = 0$. Previously, we conjectured that DALD re-equilibrates at more favourable intermediate times. Figure 3 (left) shows that DALD does recover the heavier mode weight even past the known threshold where the marginal and posterior share a log-concavity window [29, Proposition 19], whereas SI does not. Eventually, all samples collapse onto the lighter mode past a certain separation. Indeed, since our auxiliary variables are evolved via local moves, multimodality remains challenging in the absence of global information.

Vanishing score error. Our error analysis implicitly assumes the W_2^2 distance between the SMC empirical measure and the posterior is controlled by the particle count N. To test this assumption, we run DALD on the previous bi-modal mixture target, with varying particle counts and mode separations, and measure the score MSE and empirical W_2^2 distance against ground-truth posterior samples. Figure 3 (middle and right) shows our findings while further details are in Appendix H.5. Notably, with modest separation, we achieve a W_2^2 convergence rate that closely recapitulates the i.i.d. rate of $\mathcal{O}(N^{-2/d})$ in Fournier and Guillin [24] and an MSE rate close to the standard $\mathcal{O}(N^{-1})$ for bounded test functions [19, Section 3.5.1]. The rate degrades, however, with increasing separation. Hence, we believe our assumption of score error control is sound under modest separation and limited on well-separated targets that may require far larger particle counts than is practically feasible.

6 Discussion

In this work, we develop diffusion path sequential Monte Carlo (DPSMC), a sampler on diffusion paths using SMC as a backbone for score estimation. We further introduce control-variate schedules and motivate principled parameterisations of the diffusion path. Finally, under different instantiations of the path, we provide theoretical guarantees for DPSMC and test it on a range of benchmark targets.

Limitations and future work. Our results open several promising directions. First, our analysis would greatly benefit from a finite-particle complexity bound which, given the target sequence depends on the particle approximations, requires tools from mean-field particle analysis [21] and deserves its own careful treatment. Second, we find that pseudomarginal importance weights under the nested SMC scheme [59] can exhibit some variance. Exploring other transition kernels and SI instantiations that induce smoother, better-conditioned posteriors along the path is a promising strategy. Third, while we choose hyperparameters with care, their choice hinges on oracle (evaluation) metrics which may be unavailable. We think the normalising constant estimates can be informative for this. Finally, as our SMC subroutines rely on local moves, multimodality can only be tackled well in mild, controlled settings. Incorporating progressive sampling strategies, e.g. tempering, is a natural next step for tackling very high-dimensional, multimodal targets found in real-world settings.

Acknowledgements

J.M.Y. is supported by the Roth Scholarship provided by the Department of Mathematics, Imperial College London. P.C.E. is funded by EPSRC through the Centre for Doctoral Training in Modern Statistics and Statistical Machine Learning (StatML), grant no. EP/S023151/1. S.R.'s work is partially funded by the Deutsche Forschungsgemeinschaft (DFG) under Project-ID 318763901 – SFB1294. Lastly, J.M.Y. would like to thank Tim Wang for helpful discussions.

References

[1] Tara Akhound-Sadegh, Jarrid Rector-Brooks, Joey Bose, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcin Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, Nikolay Malkin, and Alexander Tong. Iterated denoising energy matching for sampling from Boltzmann densities. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 760–786. PMLR, 2024.

[2] Tara Akhound-Sadegh, Jungyoon Lee, Joey Bose, Valentin De Bortoli, Arnaud Doucet, Michael Bronstein, Dominique Beaini, Siamak Ravanbakhsh, Kirill Neklyudov, and Alexander Tong. Progressive inference-time annealing of diffusion models for sampling from Boltzmann densities. In *Advances in Neural Information Processing Systems*, volume 38, pages 63218–63251, 2025.

[3] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209): 1–80, 2025.

[4] Christophe Andrieu and Gareth O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

[5] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72 (3):269–342, 2010.

[6] Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d-linear convergence bounds for diffusion models via stochastic localization. In *The Twelfth International Conference on Learning Representations*, 2024.

[7] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[8] Patrick Cattiaux, Paula Cordero-Encinar, and Arnaud Guillin. Diffusion annealed langevin dynamics: a theoretical study. *arXiv preprint arXiv:2511.10406*, 2025.

[9] Omar Chehab and Anna Korba. A practical diffusion path for sampling. In *ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2024.

[10] Omar Chehab, Anna Korba, Austin J Stromme, and Adrien Vacher. Provable convergence and limitations of geometric tempering for Langevin dynamics. In *The Thirteenth International Conference on Learning Representations*, 2025.

[11] Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

[12] Hongrui Chen, Holden Lee, and Jianfeng Lu. Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 4735–4763. PMLR, 2023.

[13] Junhua Chen, Lorenz Richter, Julius Berner, Denis Blessing, Gerhard Neumann, and Anima Anandkumar. Sequential controlled Langevin diffusions. In *The Thirteenth International Conference on Learning Representations*, 2025.

[14] Wenlin Chen, Mingtian Zhang, Brooks Paige, José Miguel Hernández-Lobato, and David Barber. Diffusive Gibbs sampling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 7731–7747. PMLR, 2024.

[15] Paula Cordero-Encinar, O. Deniz Akyildiz, and Andrew B. Duncan. Non-asymptotic analysis of diffusion annealed Langevin Monte Carlo for generative modelling. *arXiv preprint arXiv:2502.09306*, 2025.

[16] Paula Cordero-Encinar, Andrew B. Duncan, Sebastian Reich, and O. Deniz Akyildiz. Sampling by averaging: A multiscale approach to score estimation. In *Advances in Neural Information Processing Systems*, volume 38, pages 81139–81186, 2025.

[17] Laurence Davies. *Contributions to Bayesian transdimensional algorithms*. PhD thesis, Queensland University of Technology, 2023.

[18] Valentin De Bortoli, Michael Hutchinson, Peter Wirnsberger, and Arnaud Doucet. Target score matching. *arXiv preprint arXiv:2402.08667*, 2024.

[19] Pierre Del Moral. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer, 2004.

[20] Pierre Del Moral. *Mean Field Simulation for Monte Carlo Integration*, volume 126 of *Monographs on Statistics and Applied Probability*. Chapman and Hall/CRC, 2013.

[21] Pierre Del Moral and Emmanuel Rio. Concentration inequalities for mean field particle models. *The Annals of Applied Probability*, 21(3):1017–1052, 2011.

[22] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo Samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436, 2006.

[23] Alain Durmus and Éric Moulines. Quantitative bounds of convergence for geometrically ergodic Markov chain in the Wasserstein distance with application to the Metropolis adjusted Langevin algorithm. *Statistics and Computing*, 25(1):5–19, 2015.

[24] Nicolas Fournier and Arnaud Guillin. On the rate of convergence in Wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3):707–738, 2015.

[25] Andrew Gelman and Xiao-Li Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, pages 163–185, 1998.

[26] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

[27] Charles J. Geyer and Elizabeth A. Thompson. Annealing Markov chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association*, 90(431): 909–920, 1995.

[28] Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer, 2004.

[29] Louis Grenioux, Maxence Noble, Marylou Gabrié, and Alain Oliviero Durmus. Stochastic localization via iterative posterior sampling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 16337–16376. PMLR, 2024.

[30] Wei Guo, Molei Tao, and Yongxin Chen. Provable benefit of annealed Langevin Monte Carlo for non-log-concave sampling. In *The Thirteenth International Conference on Learning Representations*, 2025.

[31] Jiajun He, Wenlin Chen, Mingtian Zhang, David Barber, and José Miguel Hernández-Lobato. Training neural samplers with reverse diffusive KL divergence. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 5167–5175. PMLR, 2025.

[32] Ye He, Kevin Rojas, and Molei Tao. Zeroth-order sampling methods for non-log-concave distributions: Alleviating metastability by denoising diffusion. In *Advances in Neural Information Processing Systems*, volume 37, pages 71122–71161, 2024.

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

[34] Xunpeng Huang, Hanze Dong, Yifan HAO, Yian Ma, and Tong Zhang. Reverse diffusion Monte Carlo. In *The Twelfth International Conference on Learning Representations*, 2024.

[35] Khaled Kahouli, Romuald Elie, Klaus-Robert Müller, Quentin Berthet, Oliver T Unke, and Arnaud Doucet. Control variate score matching for diffusion models. *arXiv preprint arXiv:2512.20003*, 2025.

[36] Ioannis Karatzas and Steven E. Shreve. *Brownian Motion and Stochastic Calculus*. Springer, 1991.

[37] Joohwan Ko and Tomas Geffner. Latent target score matching, with an application to simulation-based inference. In *Machine Learning and the Physical Sciences Workshop, NeurIPS*, 2025.

[38] Bálint Máté and François Fleuret. Learning interpolations between Boltzmann densities. *Transactions on Machine Learning Research*, 2023.

[39] Radford M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

[40] Radford M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, 2003.

[41] Radford M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*, 2(11):2, 2011.

[42] Mark E. J. Newman and Gerard T. Barkema. *Monte Carlo methods in statistical physics*. Clarendon Press, 1999.

[43] Maxence Noble, Louis Grenioux, Marylou Gabrié, and Alain Oliviero Durmus. Learned reference-based diffusion sampler for multi-modal distributions. In *The Thirteenth International Conference on Learning Representations*, 2025.

[44] Angus Phillips, Hai-Dang Dau, Michael John Hutchinson, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. Particle denoising diffusion sampler. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 40688–40724. PMLR, 2024.

[45] Lorenz Richter and Julius Berner. Improved sampling via learned diffusions. In *The Twelfth International Conference on Learning Representations*, 2024.

[46] 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*, pages 51724–51746. PMLR, 2025.

[47] Saeed Saremi, Ji Won Park, and Francis Bach. Chain of log-concave Markov chains. In *The Twelfth International Conference on Learning Representations*, 2024.

[48] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265. PMLR, 2015.

[49] 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.

[50] Robert H. Swendsen and Jian-Sheng Wang. Replica Monte Carlo simulation of spin glasses. *Physical Review Letters*, 57(21):2607–2609, 1986.

[51] Nicholas G. Tawn, Gareth O. Roberts, and Jeffrey S. Rosenthal. Weight-preserving simulated tempering. *Statistics and Computing*, 30(1):27–41, 2020.

[52] Bruce Van Brunt. *The calculus of variations*. Universitext. Springer, 2003.

[53] Francisco Vargas, Shreyas Padhy, Denis Blessing, and Nikolas Nüsken. Transport meets variational inference: Controlled Monte Carlo diffusions. In *The Twelfth International Conference on Learning Representations*, 2024.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[55] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2008.

[56] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

[57] Darren J. Wilkinson. *Stochastic modelling for systems biology*. Chapman and Hall/CRC, 2018.

[58] Dawn Woodard, Scott Schmidler, and Mark Huber. Sufficient conditions for torpid mixing of parallel and simulated tempering. *Electronic Journal of Probability*, 14:780–804, 2009.

[59] Luhuan Wu, Yi Han, Christian A. Andersson Naesseth, and John P. Cunningham. Reverse diffusion sequential Monte Carlo samplers. In *Advances in Neural Information Processing Systems*, volume 38, pages 69445–69459, 2025.

[60] Qinsheng Zhang and Yongxin Chen. Path integral sampler: A stochastic control approach for sampling. In *International Conference on Learning Representations*, 2022.

Organisation of the appendix

We present proofs for propositions stated in the main text as well as supplementary information like implementation details and additional experiments. The appendix is organised as follows:

A Additional Background **16**

A.1 Related Work on Diffusion-Based Neural Samplers 16

B Sequential Monte Carlo samplers **16**

B.1 Marginal Score and Density (Proof of Proposition 1) 16

B.2 Weight Update for MALA Kernel 17

C Score Identities **17**

C.1 Denoising and Target Score Identities for General Base ν 17

C.2 Control Variate Schedules 18

C.2.1 Optimal Scalar CV Schedule (Proof of Proposition 2) 20

C.2.2 Optimal Diagonal CV Schedule 21

C.2.3 Optimal Matrix CV Schedule (Proof of Proposition 3) 22

D Diffusion Path Details **24**

D.1 Diffusion Annealed Langevin Dynamics 24

D.2 Schedule Minimising Action Upper Bound 25

D.3 Stochastic Interpolants Dynamics for the Diffusion Path 27

E DPSMC Implementation **28**

F Theoretical Analysis **30**

F.1 Score Estimation Error (Proof of Proposition 4) 30

F.2 Overall Algorithm Error (Proof of Theorem 1) 31

G Numerical Experiments **32**

G.1 Methods and Hyperparameters 32

G.2 Target Distributions 35

G.3 Normalising Constant Estimation 36

H Ablations and Additional Experiments **37**

H.1 Choice of Score Identity 37

H.1.1 Benchmarking Other Score Identities 37

H.1.2 High-Dimensional Scaling of CV Schedules 38

H.2 Wall-clock Times 39

H.3 Comparison with Parallel Tempering 39

H.4 Initialisation Dependency 39

H.5 Posterior and Score Error 40

H.6 Hyperparameter Sensitivity 41

A Additional Background

A.1 Related Work on Diffusion-Based Neural Samplers

There are several works that use neural networks for sampling from target distributions while leveraging the diffusion path, most commonly through the OU process and its time-reversal. Below, we provide a brief discussion of related works that fall under this neural sampler category. We remark that our approach is *training-free* in contrast.

Some works directly train a diffusion model, leveraging the same score-matching objective without assuming access to samples beforehand. For example, Akhound-Sadegh et al. [1] estimate the time-varying score functions using self-normalised importance sampling together with TSI and use them as regression targets. On the other hand, some works temper the target to more easily draw an initial set of samples, from which a diffusion model is trained and steered to produce samples from a less tempered version of the learned distribution, and the process is repeated [2, 46].

Several works approach the problem from a variational inference (VI) perspective, whereby a forward and reverse process have parameterised drift functions, and these are optimised to minimise the divergence between their induced path measures, i.e. so they become time-reversals of each other [60, 53, 45, 13]. These can be analogously framed under a stochastic optimal control light.

Related to our work is PDDS [44] that reframes sampling from a target as sampling from the reward-tilted distribution of a diffusion model. Phillips et al. [44] define an SMC sampler along this path using twisting potentials parameterised by neural networks to more effectively guide the sampling process. To facilitate training, they consider their novel score matching loss, where they introduce a control-variate in the score identity, the choice of which can be viewed as MSI.

B Sequential Monte Carlo samplers

In this section, we show certain SMC results which we have used in the paper.

B.1 Marginal Score and Density (Proof of Proposition 1)

Proposition 1. *For any $k \geq 1$, let $Y_{0:k} \sim q_{0:k}$. If $\varphi_{k,x}$ forms a score identity as in* (8)*, then we have*

$$\nabla \log \mu_{t_k}(X_k) = \frac{\mathbb{E}[\varphi_{k,X_k}(Y_k) \prod_{p=0}^{k} G_p(Y_{p-1}, Y_p)]}{\mathbb{E}[\prod_{p=0}^{k} G_p(Y_{p-1}, Y_p)]}, \quad \mu_{t_k}(X_k) = \mathbb{E}\left[\prod_{p=0}^{k} G_p(Y_{p-1}, Y_p)\right]$$

where $G_0(y_{-1}, y_0) \coloneqq \tilde{\rho}_0(y_0)/q_0(y_0)$ and the expectations are taken with respect to the path measure induced by the forward kernels $(\mathsf{K}_p)_{p=1}^k$ and initial distribution q_0.

Proof. Expanding the product we have that

$$\begin{aligned}\prod_{p=0}^{k} G_p(y_{p-1}, y_p) &= \frac{\tilde{\rho}_0(y_0)}{q_0(y_0)} \prod_{p=1}^{k} \frac{\tilde{\rho}_p(y_p)\mathsf{L}_{p-1}(y_p, y_{p-1})}{\tilde{\rho}_{p-1}(y_{p-1})\mathsf{K}_p(y_{p-1}, y_p)} \\ &= \frac{\tilde{\rho}_k(y_k) \prod_{p=1}^{k} \mathsf{L}_{p-1}(y_p, y_{p-1})}{q_0(y_0) \prod_{p=1}^{k} \mathsf{K}_p(y_{p-1}, y_p)} = \frac{\tilde{\rho}_{0:k}(y_{0:k})}{q_{0:k}(y_{0:k})}.\end{aligned}$$

The ratio in the proposition is therefore given by

$$\begin{aligned}\frac{\mathbb{E}[\varphi_{k,x_k}(Y_k) \prod_{p=1}^{k} G_p(Y_{p-1}, Y_p)]}{\mathbb{E}[\prod_{p=1}^{k} G_p(Y_{p-1}, Y_p)]} &= \frac{\int q_{0:k}(y_{0:k}) \varphi_{k,x}(y_k) \frac{\tilde{\rho}_{0:k}(y_{0:k})}{q_{0:k}(y_{0:k})} \mathrm{d}y_{0:k}}{\int q_{0:k}(y_{0:k}) \frac{\tilde{\rho}_{0:k}(y_{0:k})}{q_{0:k}(y_{0:k})} \mathrm{d}y_{0:k}} \\ &= \frac{\int \tilde{\rho}_{0:k}(y_{0:k}) \varphi_{k,x_k}(y_k) \mathrm{d}y_{0:k}}{\int \tilde{\rho}_{0:k}(y_{0:k}) \mathrm{d}y_{0:k}} \\ &= \int \rho_{0:k}(y_{0:k}) \varphi_{k,x_k}(y_k) \mathrm{d}y_{0:k} = \int \rho_k(y_k) \varphi_{k,x_k}(y_k) \mathrm{d}y_k,\end{aligned}$$

where we have used the fact that $\rho_{0:k}(y_{0:k})$ admits $\rho_k(y_k)$ as a marginal. Since $\rho_k = \varrho_{t_k,x_k}$ and our test function $\varphi_{k,x}$ satisfies (8) for any $x \in \mathbb{R}^d$, then we can conclude that

$$\frac{\mathbb{E}[\varphi_{k,x}(Y_k)\prod_{p=1}^k G_p(Y_{p-1},Y_p)]}{\mathbb{E}[\prod_{p=1}^k G_p(Y_{p-1},Y_p)]} = \mathbb{E}_{\varrho_{t_k,x_k}}[\varphi_{k,x_k}(Y_k)] = \nabla \log \mu_{t_k}(x_k).$$

Now, we show the result for the path marginal density. We have

$$\begin{aligned}\mathbb{E}\left[\prod_{p=0}^k G_p(Y_{p-1},Y_p)\right] &= \int q_{0:k}(y_{0:k})\frac{\tilde{\rho}_{0:k}(y_{0:k})}{q_{0:k}(y_{0:k})}\mathrm{d}y_{0:k} = \int \tilde{\rho}_k(y_k)\prod_{p=1}^k \mathsf{L}_{p-1}(y_p,y_{p-1})\mathrm{d}y_{0:k} \\ &= \int \tilde{\rho}_k(y_k)\mathrm{d}y_k = \int \frac{1}{\sqrt{1-\lambda_{t_k}}^d}\nu\left(\frac{X_k-\sqrt{\lambda_{t_k}}y_k}{\sqrt{1-\lambda_{t_k}}}\right)\pi(y_k)\mathrm{d}y_k.\end{aligned}$$

With a change of variables $u = \sqrt{\lambda_{t_k}}y_k$, we hence have the result

$$\mathbb{E}\left[\prod_{p=0}^k G_p(Y_{p-1},Y_p)\right] = \int \frac{1}{\sqrt{1-\lambda_{t_k}}^d}\nu\left(\frac{X_k-u}{\sqrt{1-\lambda_{t_k}}}\right)\frac{1}{\sqrt{\lambda_{t_k}}^d}\pi\left(\frac{u}{\sqrt{\lambda_{t_k}}}\right)\mathrm{d}u = \mu_{t_k}(X_k).$$

□

B.2 Weight Update for MALA Kernel

In Section 3.2, we write our potential functions for general kernels $(\mathsf{K}_p)_{p=1}^k$ and $(\mathsf{L}_{p-1})_{p=1}^k$ as

$$G_k(y_{k-1},y_k) = \frac{\tilde{\rho}_k(y_k)\mathsf{L}_{k-1}(y_k,y_{k-1})}{\tilde{\rho}_{k-1}(y_{k-1})\mathsf{K}_k(y_{k-1},y_k)}. \tag{13}$$

As we mention in Section 3.6, we choose the forward kernel to be the MALA kernel which simplifies the weights. In order to see this, if K_k leaves ρ_k invariant and we can choose the backward kernel as its time-reversal,

$$\mathsf{L}_{k-1}(y_k,y_{k-1}) = \frac{\rho_k(y_{k-1})\mathsf{K}_k(y_{k-1},y_k)}{\rho_k(y_k)}, \tag{14}$$

then the incremental weight simplifies, which yields the familiar annealed importance sampling weights. In particular, we can write

$$\begin{aligned}G_k(y_{k-1},y_k) &= \frac{\tilde{\rho}_k(y_k)\mathsf{L}_{k-1}(y_k,y_{k-1})}{\tilde{\rho}_{k-1}(y_{k-1})\mathsf{K}_k(y_{k-1},y_k)} \\ &= \frac{\tilde{\rho}_k(y_k)}{\tilde{\rho}_{k-1}(y_{k-1})}\cdot\frac{\rho_k(y_{k-1})}{\rho_k(y_k)} = \frac{\tilde{\rho}_k(y_{k-1})}{\tilde{\rho}_{k-1}(y_{k-1})},\end{aligned} \tag{15}$$

where we used $\rho_k = \tilde{\rho}_k/Z_k$ so that $\tilde{\rho}_k(y_k)/\rho_k(y_k) = Z_k$ and $\rho_k(y_{k-1}) = \tilde{\rho}_k(y_{k-1})/Z_k$.

C Score Identities

In this section, we show how we can derive control-variate schedules for minimising the variance of score estimators under different regimes. We do so for general base and target densities.

C.1 Denoising and Target Score Identities for General Base ν

To define a general score test function with a control-variate schedule, we first present lemmas that extend DSI and TSI to general base distributions.

Lemma A.1 (Denoising Score Identity for General Base ν)**.** *For $t \in [0,1]$, it holds that*

$$\nabla \log \mu_t(x) = \mathbb{E}_{\varrho_{t,x}}\left[\frac{1}{\sqrt{1-\lambda_t}}\nabla \log \nu\left(\frac{x-\sqrt{\lambda_t}Y}{\sqrt{1-\lambda_t}}\right)\right].$$

Proof. Recall that the diffusion path marginal μ_t is defined by the convolution

$$\mu_t(x) = \int \frac{1}{\sqrt{\lambda_t(1-\lambda_t)}^d} \nu\left(\frac{x-u}{\sqrt{1-\lambda_t}}\right) \pi\left(\frac{u}{\sqrt{\lambda_t}}\right) \mathrm{d}u.$$

With a change of variables $u = \sqrt{\lambda_t} y$, we have $\mathrm{d}u = \sqrt{\lambda_t}^d \mathrm{d}y$ and

$$\mu_t(x) = \int \frac{1}{\sqrt{1-\lambda_t}^d} \nu\left(\frac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right) \pi(y) \mathrm{d}y.$$

Note that the integrand is precisely the unnormalised posterior. From here, we can solve for the score

$$\begin{aligned}
\nabla \log \mu_t(x) = \frac{\nabla \mu_t(x)}{\mu_t(x)} &= \frac{1}{\mu_t(x)} \int \nabla \left(\frac{1}{\sqrt{1-\lambda_t}^d} \nu\left(\frac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right) \pi(y) \right) \mathrm{d}y \\
&= \int \left(\frac{1}{\mu_t(x)} \frac{1}{\sqrt{1-\lambda_t}^d} \nu\left(\frac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right) \pi(y) \right) \frac{1}{\sqrt{1-\lambda_t}} \nabla \log \nu\left(\frac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right) \mathrm{d}y \\
&= \int \varrho_{t,x}(y) \frac{1}{\sqrt{1-\lambda_t}} \nabla \log \nu\left(\frac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right) \mathrm{d}y = \mathbb{E}_{\varrho_{t,x}}\left[\frac{1}{\sqrt{1-\lambda_t}} \nabla \log \nu\left(\frac{x-\sqrt{\lambda_t}Y}{\sqrt{1-\lambda_t}}\right)\right].
\end{aligned}$$

□

Lemma A.2 (Target Score Identity for General Base ν)**.** *For $t \in [0,1]$, it holds that*

$$\nabla \log \mu_t(x) = \mathbb{E}_{\varrho_{t,x}}\left[\frac{1}{\sqrt{\lambda_t}} \nabla \log \pi(Y)\right].$$

Proof. As the diffusion path marginal μ_t is a convolution, we have that

$$\mu_t(x) = \int \frac{1}{\sqrt{\lambda_t(1-\lambda_t)}^d} \nu\left(\frac{u}{\sqrt{1-\lambda_t}}\right) \pi\left(\frac{x-u}{\sqrt{\lambda_t}}\right) \mathrm{d}u.$$

Solving for the score

$$\begin{aligned}
\nabla \log \mu_t(x) = \frac{\nabla \mu_t(x)}{\mu_t(x)} &= \frac{1}{\mu_t(x)} \int \nabla \left(\frac{1}{\sqrt{\lambda_t(1-\lambda_t)}^d} \nu\left(\frac{u}{\sqrt{1-\lambda_t}}\right) \pi\left(\frac{x-u}{\sqrt{\lambda_t}}\right) \right) \mathrm{d}u \\
&= \int \left(\frac{1}{\mu_t(x)} \frac{1}{\sqrt{\lambda_t(1-\lambda_t)}^d} \nu\left(\frac{u}{\sqrt{1-\lambda_t}}\right) \pi\left(\frac{x-u}{\sqrt{\lambda_t}}\right) \right) \frac{1}{\sqrt{\lambda_t}} \nabla \log \pi\left(\frac{x-u}{\sqrt{\lambda_t}}\right) \mathrm{d}u.
\end{aligned}$$

Using a change of variables $u = x - \sqrt{\lambda_t} y$, we have $\mathrm{d}u = \left|-\sqrt{\lambda_t}^d\right| \mathrm{d}y$ and

$$\begin{aligned}
\nabla \log \mu_t(x) &= \int \left(\frac{1}{\mu_t(x)} \frac{1}{\sqrt{1-\lambda_t}^d} \nu\left(\frac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right) \pi\left(y\right) \right) \frac{1}{\sqrt{\lambda_t}} \nabla \log \pi(y) \mathrm{d}y \\
&= \int \varrho_{t,x}(y) \frac{1}{\sqrt{\lambda_t}} \nabla \log \pi(y) \mathrm{d}y = \mathbb{E}_{\varrho_{t,x}}\left[\frac{1}{\sqrt{\lambda_t}} \nabla \log \pi(Y)\right].
\end{aligned}$$

□

Proposition A.1. *As a consequence of Lemma A.1 and Lemma A.2, for $\mathbf{A} \in \mathbb{R}^{d \times d}$ and $t \in [0,1]$, it holds that*

$$\nabla \log \mu_t(x) = \mathbb{E}_{\varrho_{t,x}}\left[\frac{1}{\sqrt{1-\lambda_t}} \mathbf{A} \nabla \log \nu\left(\frac{x-\sqrt{\lambda_t}Y}{\sqrt{1-\lambda_t}}\right) + \frac{1}{\sqrt{\lambda_t}} (\mathbf{I} - \mathbf{A}) \nabla \log \pi(Y)\right].$$

Proof. The proof is immediate as a convex combination of the two identities given in Lemma A.1 and Lemma A.2. □

C.2 Control Variate Schedules

Remark A.1. The identity in Proposition A.1 can be viewed as introducing a matrix-valued control variate. To see this, starting from Lemma A.1 and the fact that $\mathbb{E}_{\varrho_{t,x}}[\nabla \log \varrho_{t,x}(Y)] = 0$, we can choose a control variate matrix $\mathbf{B} \in \mathbb{R}^{d \times d}$ to get

$$\begin{aligned}
\nabla \log \mu_t(x) &= \mathbb{E}_{\varrho_{t,x}}\left[\frac{1}{\sqrt{1-\lambda_t}} \nabla \log \nu\left(\frac{x-\sqrt{\lambda_t}Y}{\sqrt{1-\lambda_t}}\right) - \mathbf{B} \nabla \log \varrho_{t,x}(Y)\right] + \mathbf{B}\mathbf{0} \\
&= \mathbb{E}_{\varrho_{t,x}}\left[\frac{1}{\sqrt{1-\lambda_t}} \nabla \log \nu\left(\frac{x-\sqrt{\lambda_t}Y}{\sqrt{1-\lambda_t}}\right) - \mathbf{B}\left(-\frac{\sqrt{\lambda_t}}{\sqrt{1-\lambda_t}} \nabla \log \nu\left(\frac{x-\sqrt{\lambda_t}Y}{\sqrt{1-\lambda_t}}\right) + \nabla \log \pi(Y)\right)\right] \\
&= \mathbb{E}_{\varrho_{t,x}}\left[\frac{1}{\sqrt{1-\lambda_t}} (\mathbf{I} + \sqrt{\lambda_t}\mathbf{B}) \nabla \log \nu\left(\frac{x-\sqrt{\lambda_t}Y}{\sqrt{1-\lambda_t}}\right) - \frac{1}{\sqrt{\lambda_t}} (\sqrt{\lambda_t}\mathbf{B}) \nabla \log \pi(Y)\right].
\end{aligned}$$

Denoting $\mathbf{A} := \mathbf{I} + \sqrt{\lambda_t}\mathbf{B}$, we precisely have the convex combination stated above. □

Next we start our analysis from the scalar case. Recall that $\phi^{\alpha\mathbf{I}}_{t,x}(y) = \frac{\alpha}{\sqrt{1-\lambda_t}}\nabla \log \nu\left(\frac{x-\sqrt{\lambda_t}Y}{\sqrt{1-\lambda_t}}\right) + \frac{1-\alpha}{\sqrt{\lambda_t}}\nabla \log \pi(Y)$. Having introduced a control variate, we now seek the schedule α_t minimising the variance of score estimates.

Lemma A.3. *Fix $x \in \mathbb{R}^d$, the x-dependent CV schedule $\alpha^*_{t,x} \in \mathbb{R}$ minimising the variance of the score estimate is given by*

$$\alpha^*_{t,x} = \operatorname*{argmin}_{\alpha\in\mathbb{R}} \mathrm{Var}_{\varrho_{t,x}}[\phi^{\alpha\mathbf{I}}_{t,x}(Y)] = \frac{\mathbb{E}_{\varrho_{t,x}}\left[\langle \nabla \log \pi(Y), \nabla \log \varrho_{t,x}(Y)\rangle\right]}{\mathbb{E}_{\varrho_{t,x}}\left[\|\nabla \log \varrho_{t,x}(Y)\|^2\right]}.$$

Proof. Denote $s_1(x,y) := \frac{1}{\sqrt{1-\lambda_t}}\nabla \log \nu\left(\frac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right)$ and $s_2(y) := \frac{1}{\sqrt{\lambda_t}}\nabla \log \pi\left(y\right)$. The variance is given by

$$\mathrm{Var}_{\varrho_{t,x}}[\phi^{\alpha\mathbf{I}}_{t,x}(Y)] = \mathbb{E}_{\varrho_{t,x}}[\|\alpha s_1(x,Y) + (1-\alpha)s_2(Y)\|^2] - \|\nabla \log \mu_t(x)\|^2.$$

Ignoring the additive constant, we define the objective

$$\mathcal{L}^{\text{Var}}_{t,x}(\alpha) = \mathbb{E}_{\varrho_{t,x}}[\|\alpha s_1(x,Y) + (1-\alpha)s_2(Y)\|^2],$$

and want to find the minimiser $\alpha^*_{t,x} = \mathrm{argmin}_{\alpha\in\mathbb{R}}\, \mathcal{L}^{\text{Var}}_{t,x}(\alpha)$. Rearranging the objective in terms of α, we get

$$\begin{aligned}\mathcal{L}^{\text{Var}}_{t,x}(\alpha) = \alpha^2 \mathbb{E}_{\varrho_{t,x}}\left[\|s_1(x,Y) - s_2(Y)\|^2\right] \\ + \alpha \mathbb{E}_{\varrho_{t,x}}\left[2\langle s_2(Y), s_1(x,Y) - s_2(Y)\rangle\right] + \mathbb{E}_{\varrho_{t,x}}\left[\|s_2(Y)\|^2\right].\end{aligned}$$

Taking the derivative with respect to α, we have

$$\tfrac{\partial}{\partial a}\mathcal{L}^{\text{Var}}_{t,x}(\alpha) = 2\alpha \mathbb{E}_{\varrho_{t,x}}[\|s_1(x,Y) - s_2(Y)\|^2] + 2\mathbb{E}_{\varrho_{t,x}}[\|s_2(Y), s_1(x,Y) - s_2(Y)\|^2],$$

and equating it to zero yields the stationary solution

$$\alpha^*_{t,x} = \frac{\mathbb{E}_{\varrho_{t,x}}\left[\langle -s_2(Y), s_1(x,Y) - s_2(Y)\rangle\right]}{\mathbb{E}_{\varrho_{t,x}}\left[\|s_1(x,Y) - s_2(Y)\|^2\right]}.$$

As the objective has a positive second derivative everywhere

$$\begin{aligned}\tfrac{\partial^2}{\partial \alpha^2}\mathcal{L}^{\text{Var}}_{t,x}(\alpha) &= 2\mathbb{E}_{\varrho_{t,x}}\left[\|s_1(x,Y) - s_2(Y)\|^2\right] \\ &= 2\mathbb{E}_{\varrho_{t,x}}[\|\nabla \log \varrho_{t,x}(Y)\|^2] > 0,\end{aligned}$$

this is indeed a global minimiser. Rewriting $-s_1(x,y) + s_2(y) = \frac{1}{\sqrt{\lambda_t}}\nabla \log \varrho_{t,x}(y)$, we have

$$\alpha^*_{t,x} = \frac{\mathbb{E}_{\varrho_{t,x}}\left[\langle \nabla \log \pi(Y), \nabla \log \varrho_{t,x}(Y)\rangle\right]}{\mathbb{E}_{\varrho_{t,x}}\left[\|\nabla \log \varrho_{t,x}(Y)\|^2\right]}.$$

□

As we have auxiliary variables for estimating expectations with respect to $\varrho_{t,x}$ and evaluate the target score at their positions when propagating them, we can estimate the optimal schedule above at no additional cost. This schedule, however, is a function of x. Being mindful of estimation in practice, it may be beneficial to predict a single time-varying schedule to be used for all the samples. A natural choice is to minimise the expectation of the objective with respect to μ_t, i.e. find $\alpha^*_{t,x} = \mathrm{argmin}_{\alpha\in\mathbb{R}}\, \mathbb{E}_{\mu_t}[\mathcal{L}^{\text{Var}}_{t,X}(\alpha)]$.

To do this, we first show a result allowing us to rewrite joint expectations w.r.t. the path marginal and the posterior into expectations with respect to the endpoint distributions. This is made possible due to the hierarchical structure of how the samples and auxiliary variables are defined.

Lemma A.4. *For sufficiently smooth test functions ϕ, we have that*

$$\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[\langle \phi(Y), \nabla \log \varrho_{t,X}(Y)\rangle]] = \mathbb{E}_{\pi}[\langle \phi(X), \nabla \log \pi(X)\rangle], \tag{16}$$

$$\mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\left\langle \phi\left(\tfrac{X-\sqrt{\lambda_t}Y}{\sqrt{1-\lambda_t}}\right), \nabla \log \varrho_{t,X}(Y)\right\rangle\right]\right] = -\sqrt{\tfrac{\lambda_t}{1-\lambda_t}}\mathbb{E}_{\nu}[\langle \phi(X), \nabla \log \nu(X)\rangle]. \tag{17}$$

Proof. These results follow after invoking Stein's lemma and marginalising out a variable. We first show (16). Denote the LHS by Φ_1. Using Stein's lemma in the inner expectation, we have

$$\Phi_1 = \mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\langle\phi(Y), \nabla \log \varrho_{t,X}(Y)\rangle\right]\right] = \mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\langle\nabla, -\phi(Y)\rangle\right]\right].$$

As the inner expression only depends on Y, we can marginalise out X as follows

$$\begin{aligned}
\Phi_1 &= \iint \mu_t(x)\varrho_{t,x}(y)\langle\nabla, -\phi(y)\rangle \mathrm{d}y\mathrm{d}x = \iint \tfrac{1}{\sqrt{1-\lambda_t}^d}\nu\left(\tfrac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right)\pi\left(y\right)\langle\nabla, -\phi(y)\rangle \mathrm{d}x\mathrm{d}y \\
&= \int \pi(y)\langle\nabla, -\phi(y)\rangle \int \tfrac{1}{\sqrt{1-\lambda_t}^d}\nu\left(\tfrac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right)\mathrm{d}x\mathrm{d}y \\
&= \int \pi(y)\langle\nabla, -\phi(y)\rangle \mathrm{d}y = \mathbb{E}_\pi[\langle\nabla, -\phi(X)\rangle].
\end{aligned}$$

Invoking Stein's lemma once more, we precisely have

$$\Phi_1 = \mathbb{E}_\pi[\langle\phi(X), \nabla \log \pi(X)\rangle].$$

For (17), denote the LHS by Φ_2. We similarly have

$$\begin{aligned}
\Phi_2 &= \mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\left\langle\phi\left(\tfrac{X-\sqrt{\lambda_t}Y}{\sqrt{1-\lambda_t}}\right), \nabla \log \varrho_{t,X}(Y)\right\rangle\right]\right] \\
&= \mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\left\langle\nabla, -\phi\left(\tfrac{X-\sqrt{\lambda_t}Y}{\sqrt{1-\lambda_t}}\right)\right\rangle\right]\right] \\
&= \iint \mu_t(x)\varrho_{t,x}(y)\left\langle\nabla, -\phi\left(\tfrac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right)\right\rangle \mathrm{d}y\mathrm{d}x \\
&= \iint \tfrac{1}{\sqrt{1-\lambda_t}^d}\nu\left(\tfrac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right)\pi\left(y\right)\left\langle\nabla, -\phi\left(\tfrac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right)\right\rangle \mathrm{d}y\mathrm{d}x.
\end{aligned}$$

By a change of variables $y = \frac{1}{\sqrt{\lambda_t}}(x - \sqrt{1-\lambda_t}u)$, we have $\mathrm{d}y = \sqrt{\frac{1-\lambda_t}{\lambda_t}}^d \mathrm{d}u$ and, for any divergence, $\langle\nabla_y, f(y)\rangle = -\sqrt{\frac{\lambda_t}{1-\lambda_t}}\left\langle\nabla_u, f\left(\frac{1}{\sqrt{\lambda_t}}(x - \sqrt{1-\lambda_t}u)\right)\right\rangle$. This results in

$$\begin{aligned}
\Phi_2 &= \iint \nu\left(u\right)\pi\left(\tfrac{x-\sqrt{1-\lambda_t}u}{\sqrt{\lambda_t}}\right)\left(-\sqrt{\tfrac{\lambda_t}{1-\lambda_t}}\right)\langle\nabla, -\phi(u)\rangle \tfrac{1}{\sqrt{\lambda_t}^d}\mathrm{d}u\mathrm{d}x \\
&= -\sqrt{\tfrac{\lambda_t}{1-\lambda_t}}\int \nu(u)\langle\nabla, -\phi(u)\rangle \int \tfrac{1}{\sqrt{\lambda_t}^d}\pi\left(\tfrac{x-\sqrt{1-\lambda_t}u}{\sqrt{\lambda_t}}\right)\mathrm{d}x\mathrm{d}u \\
&= -\sqrt{\tfrac{\lambda_t}{1-\lambda_t}}\int \nu(u)\langle\nabla, -\phi(u)\rangle \mathrm{d}u = \sqrt{\tfrac{\lambda_t}{1-\lambda_t}}\mathbb{E}_\nu\left[\langle\nabla, \phi(X)\right].
\end{aligned}$$

Using Stein's lemma, we finally have

$$\Phi_2 = -\sqrt{\tfrac{\lambda_t}{1-\lambda_t}}\mathbb{E}_\nu[\langle\phi(X), \nabla \log \nu(X)\rangle].$$

□

Now, we minimise the variance in expectation, resulting in a position-independent schedule. It admits a simpler form, revealing a clear dependence on the endpoint distributions as well as the schedule λ_t.

C.2.1 Optimal Scalar CV Schedule (Proof of Proposition 2)

We restate the proposition for clarity and provide its proof.

Proposition 2. *The scalar CV schedule $\alpha_t^* \in \mathbb{R}$ minimising the expected variance is given by*

$$\alpha_t^* = \operatorname*{argmin}_{\alpha\in\mathbb{R}} \mathbb{E}_{X\sim\mu_t}[\mathrm{Var}_{Y\sim\varrho_{t,X}}[\phi_{t,X}^{\alpha\mathbf{I}}(Y)]] = \frac{\frac{1}{\lambda_t}\mathrm{Var}_\pi[\nabla \log \pi(X)]}{\frac{1}{1-\lambda_t}\mathrm{Var}_\nu[\nabla \log \nu(X)] + \frac{1}{\lambda_t}\mathrm{Var}_\pi[\nabla \log \pi(X)]}.$$

Proof. Denote $s_1(x,y) := \frac{1}{\sqrt{1-\lambda_t}}\nabla \log \nu\left(\frac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right)$ and $s_2(y) := \frac{1}{\sqrt{\lambda_t}}\nabla \log \pi\left(y\right)$. Similar to Lemma A.3, we can define the objective

$$\mathcal{L}_{t,x}^{\mathrm{Var}}(\alpha) = \mathbb{E}_{\varrho_{t,x}}[\|\alpha s_1(x,Y) + (1-\alpha)s_2(Y)\|^2].$$

Furthermore, we can follow the workings in Lemma A.3 to identically minimise this objective in expectation and yield

$$\alpha_t^* = \operatorname*{argmin}_{\alpha \in \mathbb{R}} \mathbb{E}_{\mu_t}\left[\mathcal{L}_{t,X}^{\text{Var}}(\alpha)\right] = \frac{\mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\langle \nabla \log \pi(Y), \nabla \log \varrho_{t,X}(Y)\rangle\right]\right]}{\mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\|\nabla \log \varrho_{t,X}(Y)\|^2\right]\right]}.$$

Using results from Lemma A.4, for test functions $\phi_1(y) = \nabla \log \pi(y)$ and $\phi_2(y) = \sqrt{\frac{\lambda_t}{1-\lambda_t}} \nabla \log \nu(y)$, we have that

$$\begin{aligned}
\mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\langle \phi_1(Y), \nabla \log \varrho_{t,X}(Y)\rangle\right]\right] &= \mathbb{E}_{\pi}[\langle \phi_1(X), \nabla \log \pi(X)\rangle] \\
&= \mathbb{E}_{\pi}[\|\nabla \log \pi(X)\|^2], \\
\mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\left\langle \phi_2\left(\tfrac{X-\sqrt{\lambda_t}Y}{\sqrt{1-\lambda_t}}\right), \nabla \log \varrho_{t,X}(Y)\right\rangle\right]\right] &= -\sqrt{\tfrac{\lambda_t}{1-\lambda_t}}\mathbb{E}_{\nu}[\langle \phi(X), \nabla \log \nu(X)\rangle] \\
&= -\tfrac{\lambda_t}{1-\lambda_t}\mathbb{E}_{\nu}[\|\nabla \log \nu(X)\|^2].
\end{aligned}$$

Since $\nabla \log \varrho_{t,x}(y) = -\phi_2\left(\frac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right) + \phi_1(y)$, it immediately follows that

$$\alpha_t^* = \frac{\mathbb{E}_{\pi}[\|\nabla \log \pi(X)\|^2]}{\frac{\lambda_t}{1-\lambda_t}\mathbb{E}_{\nu}\left[\|\nabla \log \nu(X)\|^2\right] + \mathbb{E}_{\pi}[\|\nabla \log \pi(X)\|^2]}.$$

Rewriting to reflect the path structure and noting $\text{Var}_p[\nabla \log p(X)] = \mathbb{E}_p[\|\nabla \log p(X)\|^2]$, we have

$$\alpha_t^* = \frac{\frac{1}{\lambda_t}\text{Var}_{\pi}[\nabla \log \pi(X)]}{\frac{1}{1-\lambda_t}\text{Var}_{\nu}[\nabla \log \nu(X)] + \frac{1}{\lambda_t}\text{Var}[\nabla \log \pi(X)]}.$$

□

A key advantage of estimating such a schedule is we can combine all our auxiliary variables across several samples to yield a larger sample size and a more stable schedule altogether. In practice, we can use the empirical distribution formed by our propagated samples in place of μ_t.

C.2.2 Optimal Diagonal CV Schedule

Now, we consider schedules that are restricted to be diagonal. We have the following result that describes a similar optimum.

Proposition A.2. *The diagonal CV schedule* $\mathbf{a}_t^* = (a_{t,1}^*, \ldots, a_{t,d}^*)^\top \in \mathbb{R}^d$*, minimising the total variance in expectation, i.e.*

$$\mathbf{a}_t^* = \operatorname*{argmin}_{\mathbf{a} \in \mathbb{R}^d} \mathbb{E}_{X \sim \mu_t}[\text{Var}_{Y \sim \varrho_{t,X}}[\phi_{t,X}^{\text{diag}(\mathbf{a})}(Y)]],$$

has its i*th component equal to*

$$(a_t^*)_i = \frac{\frac{1}{\lambda_t}\mathbb{E}_{\pi}[(\nabla \log \pi(X))_i^2]}{\frac{1}{1-\lambda_t}\mathbb{E}_{\nu}[(\nabla \log \nu(X))_i^2] + \frac{1}{\lambda_t}\mathbb{E}_{\pi}[(\nabla \log \pi(X))_i^2]}.$$

Proof. Take $\mathbf{a} = (a_1, \ldots, a_d)^\top \in \mathbb{R}^d$. Denote $s_1(x,y) := \frac{1}{\sqrt{1-\lambda_t}} \nabla \log \nu\left(\frac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right)$ and $s_2(y) := \frac{1}{\sqrt{\lambda_t}} \nabla \log \pi\,(y)$. Let us further denote the following, whose expectation is equivalent to the variance objective up to an additive constant

$$\mathcal{L}_t^{\text{Norm}}(x, y, \mathbf{A}) := \|\phi_{t,x}^{\mathbf{A}}(y)\|^2 = \|\mathbf{A}s_1(x,y) + (\mathbf{I} - \mathbf{A})s_2(y)\|^2.$$

Since $\mathbf{A}$ is diagonal, we can rewrite it to decouple the contributions of the diagonal components

$$\mathcal{L}_t^{\text{Norm}}(x, y, \text{diag}(\mathbf{a})) = \sum_{i=1}^{d} \left[a_i s_1^i(x,y) + (1 - a_i)s_2^i(y)\right]^2,$$

where s_1^i and s_2^i are the ith components of s_1 and s_2. Solving for a stationary solution to the original objective, we take the partial derivatives component-wise and move them inside the expectation, i.e. for the ith component,

$$\begin{aligned}
&\tfrac{\partial}{\partial a^i}\mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\mathcal{L}_t^{\text{Norm}}(X,Y,\text{diag}(\mathbf{a}))\right]\right] \\
&= \mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[2\left[a_i s_1^i(X,Y) + (1-a_i)s_2^i(Y)\right](s_1^i(X,Y) - s_2^i(Y))]] \\
&= 2a_i\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[(s_1^i(X,Y) - s_2^i(Y))^2]] + 2\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[s_2^i(Y)(s_1^i(X,Y) - s_2^i(Y))]].
\end{aligned}$$

Setting this to zero, we arrive at the stationary solution

$$(a_t^*)_i = \frac{\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[-s_2^i(Y)(s_1^i(X,Y) - s_2^i(Y))]]}{\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[(s_1^i(X,Y) - s_2^i(Y))^2]]}.$$

Define for now $\phi_1(u) = \frac{1}{\sqrt{1-\lambda_t}}(\nabla \log \nu(u))_i \mathbf{e}_i$, so we have $\phi_1\left(\frac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right) = s_1^i(x,y)\mathbf{e}_i$. Define also $\phi_2(u) = \frac{1}{\sqrt{\lambda_t}}(\nabla \log \pi(u))_i \mathbf{e}_i = s_2^i(u)\mathbf{e}_i$. Since $s_1(x,y) - s_2(y) = -\frac{1}{\sqrt{\lambda_t}}\nabla \log \varrho_{t,x}(y)$, we can invoke Lemma A.4 for test function ϕ_1 to get

$$\begin{aligned}
&\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[s_1^i(X,Y)(s_1^i(X,Y) - s_2^i(Y))]] \\
&= \mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[\langle s_1^i(X,Y)\mathbf{e}_i, s_1(X,Y) - s_2(Y)\rangle]] \\
&= -\tfrac{1}{\sqrt{\lambda_t}}\mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\left\langle \phi_1\left(\tfrac{X-\sqrt{\lambda_t}Y}{\sqrt{1-\lambda_t}}\right), \nabla \log \varrho_{t,X}(Y)\right\rangle\right]\right] \\
&= \tfrac{1}{\sqrt{\lambda_t}}\sqrt{\tfrac{\lambda_t}{1-\lambda_t}}\mathbb{E}_{\nu}[\langle \phi_1(X), \nabla \log \nu(X)\rangle] = \tfrac{1}{1-\lambda_t}\mathbb{E}_{\nu}[(\nabla \log \nu(X))_i^2],
\end{aligned}$$

and similarly for test function ϕ_2 to get

$$\begin{aligned}
&\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[s_2^i(Y)(s_1^i(X,Y) - s_2^i(Y))]] \\
&= \mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[\langle s_2^i(Y)\mathbf{e}_i, s_1(X,Y) - s_2(Y)\rangle]] = -\tfrac{1}{\sqrt{\lambda_t}}\mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\langle \phi_2(Y), \nabla \log \varrho_{t,X}(Y)\rangle\right]\right] \\
&= -\tfrac{1}{\sqrt{\lambda_t}}\mathbb{E}_{\pi}\left[\langle \phi_2(X), \nabla \log \pi(X)\rangle\right] = -\tfrac{1}{\lambda_t}\mathbb{E}_{\pi}[(\nabla \log \pi(X))_i^2].
\end{aligned}$$

Hence the ith component of this stationary solution is given by

$$(a_t^*)_i = \frac{\frac{1}{\lambda_t}\mathbb{E}_{\pi}[(\nabla \log \pi(X))_i^2]}{\frac{1}{1-\lambda_t}\mathbb{E}_{\nu}[(\nabla \log \nu(X))_i^2] + \frac{1}{\lambda_t}\mathbb{E}_{\pi}[(\nabla \log \pi(X))_i^2]}.$$

Inspecting the second-order partial derivatives, we have that

$$\begin{aligned}
\tfrac{\partial^2}{\partial a_i^2}\mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\mathcal{L}_t^{\text{Norm}}(X,Y,\text{diag}(\mathbf{a}))\right]\right] &= 2\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[(s_1^i(X,Y) - s_2^i(Y))^2]] \\
&= \tfrac{2}{1-\lambda_t}\mathbb{E}_{\nu}[(\nabla \log \nu(X))_i^2] + \tfrac{2}{\lambda_t}\mathbb{E}_{\pi}[(\nabla \log \pi(X))_i^2] > 0,
\end{aligned}$$

and $\frac{\partial^2}{\partial a_i \partial a_j}\mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\mathcal{L}_t^{\text{Norm}}(X,Y,\text{diag}(\mathbf{a}))\right]\right] = 0$ for $i \neq j$ for all $\mathbf{a} \in \mathbb{R}^d$. Hence, the Hessian is positive definite everywhere, given that it is diagonal with positive eigenvalues. We can conclude that the objective is convex and the stationary solution $\mathbf{a}_t^*$ is indeed a global minimiser. □

C.2.3 Optimal Matrix CV Schedule (Proof of Proposition 3)

Now, we consider arbitrary matrix schedules. We restate the proposition and provide its proof.

Proposition 3. *The matrix CV schedule $\mathbf{A}_t^* \in \mathbb{R}^{d\times d}$ minimising the expected variance is given by*

$$\mathbf{A}_t^* = \operatorname*{argmin}_{\mathbf{A}\in\mathbb{R}^{d\times d}} \mathbb{E}_{X\sim\mu_t}[\text{Var}_{Y\sim\varrho_{t,X}}[\phi_{t,X}^{\mathbf{A}}(Y)]] = \tfrac{1}{\lambda_t}\mathcal{I}_\pi\left(\tfrac{1}{1-\lambda_t}\mathcal{I}_\nu + \tfrac{1}{\lambda_t}\mathcal{I}_\pi\right)^{-1},$$

where $\mathcal{I}_p := \text{Cov}_p[\nabla \log p(X)] = \mathbb{E}_p[\nabla \log p(X) \nabla \log p(X)^\top]$ denotes the score covariance of p.

Proof. Take $\mathbf{A} \in \mathbb{R}^{d\times d}$ and let A_{ij} be its (i,j) entry. Denote $s_1(x,y) := \frac{1}{\sqrt{1-\lambda_t}}\nabla \log \nu\left(\frac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right)$ and $s_2(y) := \frac{1}{\sqrt{\lambda_t}}\nabla \log \pi(y)$. Let us refer to the following, whose expectation is equivalent to the variance objective up to an additive constant

$$\mathcal{L}_t^{\text{Norm}}(x,y,\mathbf{A}) := \|\phi_{t,x}^{\mathbf{A}}(y)\|^2 = \|\mathbf{A}s_1(x,y) + (\mathbf{I}-\mathbf{A})s_2(y)\|^2.$$

We can rewrite it as follows to show the contributions of the individual entries of the matrix

$$\begin{aligned}\mathcal{L}_t^{\text{Norm}}(x,y,\mathbf{A}) &= \sum_{l=1}^{d} \left(s_2(y) + \mathbf{A}\left(s_1(x,y) - s_2(y)\right)\right)_l^2 \\ &= \sum_{l=1}^{d} \left(s_2^l(y) + \sum_{k=1}^{d} A_{lk}\left(s_1^k(x,y) - s_2^k(y)\right)\right)^2,\end{aligned}$$

where s_1^i and s_2^i are the ith components of s_1 and s_2. Taking the derivative w.r.t. A_{ij}, we have

$$\begin{aligned}\tfrac{\partial}{\partial A_{ij}}\mathcal{L}_t^{\text{Norm}}(x,y,\mathbf{A}) &= 2\left(s_2^i(y) + \sum_{k=1}^{d} A_{ik}(s_1^k(x,y) - s_2^k(y))\right)(s_1^j(x,y) - s_2^j(y)) \\ &= 2s_2^i(y)(s_1^j(x,y) - s_2^j(y)) \\ &\quad + 2\sum_{k=1}^{d} A_{ik}(s_1^k(x,y) - s_2^k(y))(s_1^j(x,y) - s_2^j(y)).\end{aligned}$$

Now, we can consider the actual objective and take the derivative w.r.t. A_{ij}

$$\begin{aligned}\tfrac{\partial}{\partial A_{ij}}\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[\mathcal{L}_t^{\text{Norm}}(X,Y,\mathbf{A})]] &= \mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\tfrac{\partial}{\partial A_{ij}}\mathcal{L}_t^{\text{Norm}}(X,Y,\mathbf{A})\right]\right] \\ &= 2\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[s_2^i(Y)(s_1^j(X,Y) - s_2^j(Y))]] \\ &\quad + 2\sum_{k=1}^{d} A_{ik}\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[s_1^k(X,Y) - s_2^k(Y))(s_1^j(X,Y) - s_2^j(Y))]].\end{aligned}$$

Define $\phi_1(u) = \frac{1}{\sqrt{1-\lambda_t}}(\nabla \log \nu(u))_i \mathbf{e}_j$, so we have $\phi_1\left(\frac{x-\sqrt{\lambda_t}y}{\sqrt{1-\lambda_t}}\right) = s_1^i(x,y)\mathbf{e}_j$, and also $\phi_2(u) = \frac{1}{\sqrt{\lambda_t}}(\nabla \log \pi(u))_i \mathbf{e}_j = s_2^i(u)\mathbf{e}_j$. Using the fact that $s_1(x,y) - s_2(y) = -\frac{1}{\sqrt{\lambda_t}}\nabla \log \varrho_{t,x}(y)$, we can invoke Lemma A.4 for test function ϕ_1 to get

$$\begin{aligned}&\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[s_1^i(X,Y)(s_1^j(X,Y) - s_2^j(Y))]] \\ &= \mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[\langle s_1^i(X,Y)\mathbf{e}_j, s_1(X,Y) - s_2(Y)\rangle]] \\ &= -\tfrac{1}{\sqrt{\lambda_t}}\mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\left\langle \phi_1\left(\tfrac{X-\sqrt{\lambda_t}Y}{\sqrt{1-\lambda_t}}\right), \nabla \log \varrho_{t,X}(Y)\right\rangle\right]\right] \\ &= \tfrac{1}{\sqrt{\lambda_t}}\sqrt{\tfrac{\lambda_t}{1-\lambda_t}}\mathbb{E}_{\nu}[\langle \phi_1(X), \nabla \log \nu(X)\rangle] = \tfrac{1}{1-\lambda_t}\mathbb{E}_{\nu}[(\nabla \log \nu(X))_i(\nabla \log \nu(X))_j],\end{aligned}$$

and similarly for test function ϕ_2 to get

$$\begin{aligned}&\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[s_2^i(Y)(s_1^j(X,Y) - s_2^j(Y))]] \\ &= \mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[\langle s_2^i(Y)\mathbf{e}_j, s_1(X,Y) - s_2(Y)\rangle]] = -\tfrac{1}{\sqrt{\lambda_t}}\mathbb{E}_{\mu_t}\left[\mathbb{E}_{\varrho_{t,X}}\left[\langle \phi_2(Y), \nabla \log \varrho_{t,X}(Y)\rangle\right]\right] \\ &= -\tfrac{1}{\sqrt{\lambda_t}}\mathbb{E}_{\pi}\left[\langle \phi_2(X), \nabla \log \pi(X)\rangle\right] = -\tfrac{1}{\lambda_t}\mathbb{E}_{\pi}[(\nabla \log \pi(X))_i(\nabla \log \pi(X))_j].\end{aligned}$$

Note that we can rewrite

$$\mathbb{E}_p[(\nabla \log p(X))_i(\nabla \log p(X))_j] = (\mathbb{E}_p[\nabla \log p(X)\nabla \log p(X)^\top])_{ij} = (\mathcal{I}_p)_{ij},$$

and we have the simplified equalities

$$\begin{aligned}\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[s_1^i(X,Y)(s_j^j(X,Y) - s_2^j(Y))]] &= \tfrac{1}{1-\lambda_t}(\mathcal{I}_\nu)_{ij}, \\ \mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[s_2^i(Y)(s_1^j(X,Y) - s_2^j(Y))]] &= -\tfrac{1}{\lambda_t}(\mathcal{I}_\pi)_{ij}.\end{aligned}$$

We therefore have

$$\begin{aligned}\tfrac{\partial}{\partial A_{ij}}\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[\mathcal{L}_t^{\text{Norm}}(X,Y,\mathbf{A})]] &= -\tfrac{2}{\lambda_t}(\mathcal{I}_\pi)_{ij} + 2\sum_{k=1}^{d} A_{ik}\left(\tfrac{1}{1-\lambda_t}(\mathcal{I}_\nu)_{kj} + \tfrac{1}{\lambda_t}(\mathcal{I}_\pi)_{kj}\right) \\ &= -\tfrac{2}{\lambda_t}(\mathcal{I}_\pi)_{ij} + 2\left(\mathbf{A}\left(\tfrac{1}{1-\lambda_t}\mathcal{I}_\nu + \tfrac{1}{\lambda_t}\mathcal{I}_\pi\right)\right)_{ij}.\end{aligned}$$

Setting this to zero, the stationary solution must satisfy

$$\left(\mathbf{A}_t^* \left(\tfrac{1}{1-\lambda_t}\mathcal{I}_\nu + \tfrac{1}{\lambda_t}\mathcal{I}_\pi\right)\right)_{ij} = \left(\tfrac{1}{\lambda_t}\mathcal{I}_\pi\right)_{ij}.$$

Since this holds for all entries (i, j), we therefore have that

$$\mathbf{A}_t^* \left(\tfrac{1}{1-\lambda_t}\mathcal{I}_\nu + \tfrac{1}{\lambda_t}\mathcal{I}_\pi\right) = \tfrac{1}{\lambda_t}\mathcal{I}_\pi \iff \mathbf{A}_t^* = \tfrac{1}{\lambda_t}\mathcal{I}_\pi \left(\tfrac{1}{1-\lambda_t}\mathcal{I}_\nu + \tfrac{1}{\lambda_t}\mathcal{I}_\pi\right)^{-1}. \tag{18}$$

Inspecting the second-order partial derivatives, we have

$$\begin{aligned}\tfrac{\partial^2}{\partial A_{ij}\partial A_{ik}}\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[\mathcal{L}_t^{\text{Norm}}(X, Y, \mathbf{A})]] &= 2\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[s_1^k(X, Y) - s_2^k(Y))(s_1^j(X, Y) - s_2^j(Y)]] \\ &= \tfrac{2}{1-\lambda_t}(\mathcal{I}_\nu)_{kj} + \tfrac{2}{\lambda_t}(\mathcal{I}_\pi)_{kj} = 2\left(\tfrac{1}{1-\lambda_t}\mathcal{I}_\nu + \tfrac{1}{\lambda_t}\mathcal{I}_\pi\right)_{kj},\end{aligned}$$

and $\tfrac{\partial^2}{\partial A_{ij}\partial A_{lk}}\mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[\mathcal{L}_t^{\text{Norm}}(X, Y, \mathbf{A})]] = 0$ when $i \neq l$. Let us denote by $\mathbf{a} = (a_1, \ldots, a_{d^2})^\top \in \mathbb{R}^{d^2}$, the flattened representation of $\mathbf{A}$, with $a_{d(i-1)+j} = A_{ij}$. It follows then that our Hessian is a block diagonal matrix

$$\mathbf{H}(\mathbf{a}) = \left(\frac{\partial^2 \mathbb{E}_{\mu_t}[\mathbb{E}_{\varrho_{t,X}}[\mathcal{L}_t^{\text{Norm}}(X, Y, \mathbf{A})]]}{\partial a_i \partial a_j}\right) = \begin{bmatrix} \tilde{\mathbf{H}} & 0 & 0 & 0 \\ 0 & \tilde{\mathbf{H}} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \tilde{\mathbf{H}} \end{bmatrix},$$

where the identical blocks are given by

$$\tilde{\mathbf{H}} = \tfrac{2}{1-\lambda_t}\mathcal{I}_\nu + \tfrac{2}{\lambda_t}\mathcal{I}_\pi.$$

As the score covariances are positive definite, their linear combination in $\tilde{\mathbf{H}}$ is also positive definite and has positive eigenvalues. The block diagonal $\mathbf{H}(\mathbf{a})$ inherits the eigenvalues of its blocks, which are all positive, and hence the Hessian is positive definite everywhere. We can conclude that our objective is convex and the stationary solution $\mathbf{A}_t^*$ is indeed the global minimiser. □

D Diffusion Path Details

In this section, we point out the biasedness of DALD, how we can use this to inform our choice for the λ_t schedule, and stochastic interpolants dynamics for inducing the diffusion path.

D.1 Diffusion Annealed Langevin Dynamics

It is worth noting that, even when DALD (3) is simulated without discretisation error, it remains biased, since the marginal distributions of the solution to the corresponding SDE do not exactly coincide with μ_t, in contrast to the stochastic interpolants formulation proposed by Albergo et al. [3]. However, this bias can be quantified. In particular, bounds have been established in Guo et al. [30, Theorem 1] and Cordero-Encinar et al. [15, Theorem A.3]. We state the result below.

Theorem A.1. *Let $\mathbb{P}_{\text{DALD}} = (p_{t,\text{DALD}})_{t\in[0,1]}$ be the path measure of the diffusion annealed Langevin dynamics* (3)*, and $\mathbb{P} = (\mu_t)_{t\in[0,1]}$ that of a reference SDE such that the marginals at each time have distribution μ_t. If $p_{0,\text{DALD}} = \mu_0$, the KL divergence between the path measures is given by*

$$\text{KL}(\hat{\mu}_T \parallel p_{T,\text{DALD}}) \leq \text{KL}(\mathbb{P} \parallel \mathbb{P}_{\text{DALD}}) = \frac{\epsilon}{4}\mathcal{A}(\mu).$$

The proof follows from an application of Girsanov's theorem [36], we refer to the aforementioned works for details. Following Cordero-Encinar et al. [15, Lemma 3.3], we can obtain an upper bound for the action of the diffusion path (1). We outline the details below.

First, consider the reparametrised version of μ_t in terms of the schedule λ_t, denoted as $\tilde{\mu}_\lambda$ and let $X_\lambda \sim \tilde{\mu}_\lambda$ and $X_{\lambda+\delta} \sim \tilde{\mu}_{\lambda+\delta}$. Recall that

$$X_\lambda = \sqrt{\lambda}X + \sqrt{1-\lambda}\sigma Z \tag{19}$$

where $X \sim \pi$ and $Z \sim \mathcal{N}(0, \mathbf{I})$ are independent random variables. The Wasserstein-2 distance between $\tilde{\mu}_\lambda$ and $\tilde{\mu}_{\lambda+\delta}$ is given by

$$\begin{aligned} W_2^2(\tilde{\mu}_\lambda, \tilde{\mu}_{\lambda+\delta}) &\leq \mathbb{E}\left[\|X_\lambda - X_{\lambda+\delta}\|^2\right] \\ &= \mathbb{E}\left[\left\|(\sqrt{\lambda} - \sqrt{\lambda+\delta})X\right\|^2\right] + \mathbb{E}\left[\left\|\left(\sqrt{1-\lambda} - \sqrt{1-\lambda-\delta}\right)\sigma Z\right\|^2\right] \\ &= (\sqrt{\lambda} - \sqrt{\lambda+\delta})^2 \mathbb{E}\left[\|X\|^2\right] + \left(\sqrt{1-\lambda} - \sqrt{1-\lambda-\delta}\right)^2 \sigma^2 d. \end{aligned}$$

Using the definition of the metric derivative we have

$$\left|\dot{\tilde{\mu}}\right|_\lambda^2 = \lim_{\delta \to 0} \frac{W_2^2(\tilde{\mu}_\lambda, \tilde{\mu}_{\lambda+\delta})}{\delta^2} \leq \frac{\mathbb{E}[\|X\|^2]}{4\lambda} + \frac{\sigma^2 d}{4(1-\lambda)}.$$

Therefore, we have the following expression for the action

$$\begin{aligned} \mathcal{A}_\lambda(\mu) &= \int_0^1 |\dot{\mu}|_t^2 \mathrm{d}t = \int_0^1 \left|\dot{\tilde{\mu}}\right|_\lambda^2 \left|\dot{\lambda}_t\right|^2 \mathrm{d}t \\ &\leq \int_0^1 \left(\frac{\mathbb{E}[\|X\|^2]}{4\lambda_t} + \frac{\sigma^2 d}{4(1-\lambda_t)}\right) \dot{\lambda}_t^2 \mathrm{d}t \end{aligned}$$

This expression will be used later to derive a principled schedule λ_t that minimises this upper bound.

D.2 Schedule Minimising Action Upper Bound

We want to find the curve $\lambda = (\lambda_t)_{t \in [0,1]}$ that minimises the upper bound of the action functional. In Theorem A.1, we were able to bound the action by a simpler functional. The following result indicates its minimiser.

Proposition A.3. *The cosine schedule $\lambda_t = \sin^2\left(\frac{\pi t}{2}\right)$ is the unique minimiser of the functional*

$$J[\lambda] = \int_0^1 \left(\frac{\sigma^2 d}{4(1-\lambda_t)} + \frac{\mathbb{E}_\pi[\|X\|^2]}{4\lambda_t}\right) \dot{\lambda}_t^2 \mathrm{d}t.$$

Proof. Define the Lagrangian as

$$L(t, \lambda_t, \dot{\lambda}_t) = \dot{\lambda}_t^2 \left(\frac{\sigma^2 d}{4(1-\lambda_t)} + \frac{\mathbb{E}_\pi[\|X\|^2]}{4\lambda_t}\right),$$

then any stationary points of the functional $J[\lambda] = \int_0^1 L(t, \lambda_t, \dot{\lambda}_t) dt$ must satisfy the Euler-Lagrange equation

$$\frac{\partial L}{\partial \lambda_t} - \frac{d}{dt}\frac{\partial L}{\partial \dot{\lambda}_t} = 0.$$

We drop the time index in writing λ and $\dot{\lambda}$ for conciseness. We have that

$$\begin{aligned} \frac{\partial L}{\partial \lambda} &= \dot{\lambda}^2 \left(-\frac{\mathbb{E}_\pi[\|X\|^2]}{4\lambda^2} + \frac{\sigma^2 d}{4(1-\lambda)^2}\right) \\ -\frac{d}{dt}\frac{\partial L}{\partial \dot{\lambda}} &= -\frac{d}{dt}\left(2\dot{\lambda}\left(\frac{\mathbb{E}_\pi[\|X\|^2]}{4\lambda} + \frac{\sigma^2 d}{4(1-\lambda)}\right)\right) \\ &= -2\ddot{\lambda}\left(\frac{\mathbb{E}_\pi[\|X\|^2]}{4\lambda} + \frac{\sigma^2 d}{4(1-\lambda)}\right) - 2\dot{\lambda}^2\left(-\frac{\mathbb{E}_\pi[\|X\|^2]}{4\lambda^2} + \frac{\sigma^2 d}{4(1-\lambda)^2}\right). \end{aligned}$$

Equating their sum to zero, we can solve for $\ddot{\lambda}$

$$\begin{aligned} \ddot{\lambda} &= -\frac{1}{2}\dot{\lambda}^2 \frac{-\frac{\mathbb{E}_\pi[\|X\|^2]}{4\lambda^2} + \frac{\sigma^2 d}{4(1-\lambda)^2}}{\frac{\mathbb{E}_\pi[\|X\|^2]}{4\lambda} + \frac{\sigma^2 d}{4(1-\lambda)}} \\ &= \frac{1}{2}\dot{\lambda}^2 \frac{(1-\lambda)^2 \mathbb{E}_\pi\left[\|X\|^2\right] - \lambda^2 \sigma^2 d}{\lambda(1-\lambda)^2 \mathbb{E}_\pi\left[\|X\|^2\right] + \lambda^2(1-\lambda)\sigma^2 d}. \end{aligned}$$

We denote the ratio $\sigma^2 d/\mathbb{E}_\pi\left[\|X\|^2\right] =: R$ and rewrite the above as

$$\ddot{\lambda} = \frac{1}{2}\dot{\lambda}^2 \frac{(1-\lambda)^2 - R\lambda^2}{\lambda(1-\lambda)^2 + R\lambda^2(1-\lambda)},$$

As there are no t terms, we can use the substitution $\dot{\lambda} = v(\lambda)$, which yields $\ddot{\lambda} = \dot{v}v$. We have

$$\dot{v}(\lambda)v(\lambda) = \frac{1}{2}v(\lambda)^2 \frac{(1-\lambda)^2 - R\lambda^2}{\lambda(1-\lambda)^2 + R\lambda^2(1-\lambda)},$$

and rearranging reveals a first-order differential equation

$$\dot{v}(\lambda) + v(\lambda) \underbrace{\left[-\frac{(1-\lambda)^2 - R\lambda^2}{2\lambda(1-\lambda)^2 + 2R\lambda^2(1-\lambda)}\right]}_{p(\lambda)} = 0.$$

which we can solve using an integrating factor

$$\begin{aligned} I(\lambda) &= \exp\left(\int p(\lambda)\mathrm{d}\lambda\right) = \exp\left(\int \frac{R\lambda^2 - (1-\lambda)^2}{2\lambda(1-\lambda)((R-1)\lambda+1)}\mathrm{d}\lambda\right) \\ &= \exp\left(\int \frac{1}{2}\left(\frac{R-1}{(R-1)\lambda+1} - \frac{1}{\lambda} + \frac{1}{1-\lambda}\right)\mathrm{d}\lambda\right) \\ &= \exp\left(\frac{1}{2}(\log|(R-1)\lambda+1| - \log|\lambda| - \log|1-\lambda|)\right) = \sqrt{\frac{|(R-1)\lambda+1|}{\lambda(1-\lambda)}}. \end{aligned}$$

This leads to

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(v(\lambda)I(\lambda)\right) = 0 \iff v(\lambda)I(\lambda) = c_1.$$

Since $\dot{\lambda} = v(\lambda)$, the above is a separable first-order differential equation

$$\begin{aligned} \frac{\mathrm{d}\lambda}{\mathrm{d}t} I(\lambda) = c_1 &\iff \int I(\lambda)\mathrm{d}\lambda = \int c_1 \mathrm{d}t \\ &\int \sqrt{\frac{|(R-1)\lambda+1|}{\lambda(1-\lambda)}}\mathrm{d}\lambda = c_1 t + c_2 \end{aligned}$$

For arbitrary $R > 0$, the above integral does not appear to have a closed-form solution. Crucially, however, when $R = 1$, i.e. $\sigma^2 = \mathbb{E}_\pi\left[\|X\|^2\right]/d$, then we have

$$c_1 t + c_2 = \int \sqrt{\frac{1}{\lambda(1-\lambda)}}\mathrm{d}\lambda = \arcsin(2\lambda - 1).$$

The boundary conditions, $\lambda_0 = 0$ and $\lambda_1 = 1$, yield $c_2 = -\pi/2$ and $c_1 1 + c_2 = \pi/2$, respectively, so $c_1 = \pi$ and we arrive at the solution

$$\lambda_t = \frac{1 + \sin\left(\pi t - \frac{\pi}{2}\right)}{2} = \frac{1 - \cos(\pi t)}{2} = \sin^2\left(\frac{\pi t}{2}\right).$$

To conclude, we show that the functional $J[\lambda]$ is convex, which should imply the stationary point we found is indeed a global minimiser. We check below that for any $\lambda_1, \lambda_2 \in C^1[0,1]$ and any $\alpha \in [0,1]$, it holds

$$J[\alpha\lambda_1 + (1-\alpha)\lambda_2] \leq \alpha J[\lambda_1] + (1-\alpha)J[\lambda_2].$$

For simplicity, we remove the constants and work with

$$J[\lambda] = \int_0^1 \frac{\dot{\lambda}_t^2}{\lambda_t(1-\lambda_t)}\mathrm{d}t.$$

Consider the multivariate function $L(\lambda, \dot{\lambda}) = \frac{\dot{\lambda}^2}{\lambda(1-\lambda)}$, where $\lambda \in [0,1]$ and $\dot{\lambda} \in \mathbb{R}$. The Hessian of this function is given by

$$\nabla^2 L = 2 \begin{pmatrix} \dot{\lambda}^2 \frac{\lambda(1-\lambda)+(1-2\lambda)^2}{\lambda^3(1-\lambda)^3} & -\dot{\lambda}\frac{1-2\lambda}{\lambda^2(1-\lambda)^2} \\ -\dot{\lambda}\frac{1-2\lambda}{\lambda^2(1-\lambda)^2} & \frac{1}{\lambda(1-\lambda)} \end{pmatrix}.$$

It is immediate to check that the determinant is

$$\det \nabla^2 L = \frac{4\dot{\lambda}^2}{\lambda^3(1-\lambda)^3} \geq 0,$$

and $(\nabla^2 L)_{22} \geq 0$, therefore, $\nabla^2 L$ is positive semidefinite and hence $L(\lambda, \dot{\lambda})$ is jointly convex in $(\lambda, \dot{\lambda})$. Using Van Brunt [52, Theorem 10.7.1] this leads to

$$\begin{aligned} J[\alpha\lambda_1 + (1-\alpha)\lambda_2] &= \int_0^1 L(\alpha\lambda_1 + (1-\alpha)\lambda_2, \alpha\dot{\lambda}_1 + (1-\alpha)\dot{\lambda}_2)\,\mathrm{d}t \\ &\leq \int_0^1 \alpha L(\lambda_1, \dot{\lambda}_1) + (1-\alpha)L(\lambda_2, \dot{\lambda}_2)\,\mathrm{d}t \\ &= \alpha J[\lambda_1] + (1-\alpha)J[\lambda_2], \end{aligned}$$

which concludes that $J[\lambda]$ is convex and hence $\lambda_t = \sin^2\left(\frac{\pi t}{2}\right)$ is a minimiser. □

In fact, we now have the following upper bound for the action

Corollary A.1. *Under the choice $\sigma^2 = \mathbb{E}_\pi[\|X\|^2]/d$ and the cosine schedule $\lambda_t = \sin^2\left(\frac{\pi t}{2}\right)$, the action is bounded by*

$$\mathcal{A}(\mu) \leq \frac{\pi^2}{4}\mathbb{E}_\pi\left[\|X\|^2\right].$$

D.3 Stochastic Interpolants Dynamics for the Diffusion Path

Stochastic interpolants [3] are a general framework for characterising flow- and diffusion-based methods. We follow their framework and consider the stochastic interpolant $X_t = I(t, X_0, X_1)$ corresponding to the same process discussed in the main text, i.e.

$$I(t, X_0, X_1) = \sqrt{1-\lambda_t}X_0 + \sqrt{\lambda_t}X_1,$$

with boundary conditions $\lambda_0 = 0$ and $\lambda_1 = 1$ and random variables $X_0 \sim \nu$ and $X_1 \sim \pi$ independent. As such, $\mathrm{Law}(X_t) = \mu_t$, and we are interested in dynamics that induces this stochastic process. From Albergo et al. [3, Theorem 6], X_t has a density w.r.t. Lebesgue that satisfies the continuity equation

$$\partial_t \mu_t(x) + \langle \nabla, \mu_t(x) b_t(x) \rangle = 0,$$

where $b_t : \mathbb{R}^d \to \mathbb{R}^d$ is the velocity defined by

$$b_t(x) = \mathbb{E}[\partial_t I(t, X_0, X_1) \mid X_t = x].$$

We have the time-derivative

$$\partial_t I(t, X_0, X_1) = -\frac{\dot{\lambda}_t}{2\sqrt{1-\lambda_t}}X_0 + \frac{\dot{\lambda}_t}{2\sqrt{\lambda_t}}X_1,$$

and, hence, conditioned on $X_t = x$,

$$\begin{aligned} b_t(x) &= \frac{\int \left[-\frac{\dot{\lambda}_t}{2\sqrt{1-\lambda_t}}\left(\frac{x-\sqrt{\lambda_t}x_1}{\sqrt{1-\lambda_t}}\right) + \frac{\dot{\lambda}_t}{2\sqrt{\lambda_t}}x_1\right]\nu\left(\frac{x-\sqrt{\lambda_t}x_1}{\sqrt{1-\lambda_t}}\right)\pi(x_1)\mathrm{d}x_1}{\int \nu\left(\frac{x-\sqrt{\lambda_t}x_1}{\sqrt{1-\lambda_t}}\right)\pi(x_1)\mathrm{d}x_1} \\ &= \frac{\dot{\lambda}_t}{2}\int\left(\frac{\sqrt{\lambda_t}x_1 - x}{1-\lambda_t} + \frac{x_1}{\sqrt{\lambda_t}}\right)\varrho_{t,x}(x_1)\mathrm{d}x_1 = \frac{\dot{\lambda}_t}{\lambda_t}\int\left(\frac{\sqrt{\lambda_t}x_1 - x}{1-\lambda_t} + \frac{x_1/\sqrt{\lambda_t} - \sqrt{\lambda_t}x_1}{1-\lambda_t}\right)\varrho_{t,x}(x_1)\mathrm{d}x_1 \\ &= \frac{\dot{\lambda}_t}{2\lambda_t}\int\left(x + \frac{\sqrt{\lambda_t}x_1 - x}{1-\lambda_t}\right)\varrho_{t,x}(x_1)\mathrm{d}x_1 = \frac{\dot{\lambda}_t}{2\lambda_t}\left(x + \sigma^2\nabla\log\mu_t(x)\right), \end{aligned}$$

where we used DSI at the end. Furthermore, we have, for any $\gamma_t \geq 0$,

$$\partial_t \mu_t(x) = -\langle \nabla, \mu_t(x)(b_t(x) + \gamma_t \nabla \log \mu_t(x)) \rangle + \gamma_t \Delta \mu_t(x).$$

The above Fokker-Planck equation precisely describes the density evolution of the stochastic process $(X_t^{\mathrm{F}})_{t\in[0,1]}$ governed by the SDE

$$\mathrm{d}X_t^{\mathrm{F}} = \left[\tfrac{\dot{\lambda}_t}{2\lambda_t} X_t^{\mathrm{F}} + \left(\sigma^2 \tfrac{\dot{\lambda}_t}{2\lambda_t} + \gamma_t\right) \nabla \log \mu_t(X_t^{\mathrm{F}})\right] \mathrm{d}t + \sqrt{2\gamma_t}\mathrm{d}W_t, \quad X_0^{\mathrm{F}} \sim \nu.$$

With X_t and X_t^{F} coinciding in their marginal law, we indeed have that (2) induces the diffusion path.

E DPSMC Implementation

In this section, we briefly outline how DPSMC is implemented in practice. Our base algorithm in Algorithm 1 is presented as a generic, idealised version with non-interacting particles. However, as we do not know certain quantities, such as the target score covariance and the right step size for auxiliary variables, we estimate them using our ensemble of samples and auxiliary variables. Algorithm 2 depicts the scheme we use in practice. Below we detail our sampler choices.

MALA proposal. Note that $\mathsf{K}_k^{\mathrm{MALA}}$ is ρ_k-invariant, and under a suitable backwards kernel choice L_{k-1} as in (14), we have a simplified weight update given by the potential

$$G_k(y_{k-1}, y_k) = \frac{\tilde{\varrho}_{t_k, X_k}(y_{k-1})}{\tilde{\varrho}_{t_{k-1}, X_{k-1}}(y_{k-1})} = \frac{\mathcal{N}(\sqrt{\lambda_{t_k}} y_{k-1}; X_k, \sigma^2(1-\lambda_{t_k})\mathbf{I})}{\mathcal{N}(\sqrt{\lambda_{t_{k-1}}} y_{k-1}; X_{k-1}, \sigma^2(1-\lambda_{t_{k-1}})\mathbf{I})}.$$

For MALA, we scale the step size geometrically (i.e. by a factor of 1.1) to maintain an acceptance ratio of 75%. We choose a larger scaling factor in contrast to existing works using MCMC as we only do a single MALA step per iteration. This is particularly important to allow the proposal to adapt well to the rapidly changing posterior. During LL, we use a larger scale factor of 2.0 to quickly alleviate poor initial step size choices. The MALA proposal is presented in Algorithm 3.

Algorithm 3 MALA Kernel

input auxiliary variable Y_{k-1}, sample X_k, step size h_{k-1} (adjusted automatically)

1: $q_k(y' \mid y) := \mathcal{N}(y'; y + h_{k-1}\nabla \log \varrho_{t_k, X_k}(y), 2h_{k-1}\mathbf{I})$

2: $\tilde{Y}_k \sim q_k(\cdot \mid Y_{k-1})$

3: $\alpha = \min\left\{1, \frac{\tilde{\varrho}_{t_k, X_k}(\tilde{Y}_k) q_k(Y_{k-1} \mid \tilde{Y}_k)}{\tilde{\varrho}_{t_k, X_k}(Y_{k-1}) q_k(\tilde{Y}_k \mid Y_{k-1})}\right\}$

4: $Y_k \sim \alpha \delta_{\tilde{Y}_k}(\cdot) + (1-\alpha)\delta_{Y_{k-1}}(\cdot)$

output Y_k

Control variate schedule estimation. We use MMCVSI for score estimation. Algorithm 4 describes the process of estimating the matrix schedule using our samples, including the scalar and diagonal variants for completeness. Notably, since the target score covariance is a static unknown, we can reuse estimates from previous time-steps. We maintain a running average in practice. While a matrix inversion is performed, we do not perform any additional matrix regularisation or pseudoinverse. We believe the stability of our matrix schedule is aided by a form of Tikhonov regularisation resulting from the Gaussian score covariance (see line 4 in Algorithm 4).

Algorithm 4 CV Schedule Estimation

input weights $\{w_k^{i,j}\}$, auxiliary variables $\{Y_k^{i,j}\}$, samples $\{X_k^i\}$

1: $\hat{\mathcal{I}}_\pi = \frac{1}{N_x} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} w_k^{i,j} \nabla \log \pi(Y_k^{i,j}) \nabla \log \varrho_{t_k, X_k^i}(Y_k^{i,j})^\top$

2: $\alpha_{t_k} = \frac{(1-\lambda_{t_k})\mathrm{Tr}(\hat{\mathcal{I}}_\pi)}{\lambda_{t_k} d/\sigma^2 + (1-\lambda_{t_k})\mathrm{Tr}(\hat{\mathcal{I}}_\pi)}$

3: $\mathbf{a}_{t_k} = \mathrm{diag}\left(\left\{\frac{(1-\lambda_{t_k})(\hat{\mathcal{I}}_\pi)_{ii}}{\lambda_{t_k}/\sigma^2 + (1-\lambda_{t_k})(\hat{\mathcal{I}}_\pi)_{ii}}\right\}_{i=1}^d\right)$

4: $\mathbf{A}_{t_k} = \hat{\mathcal{I}}_\pi \left(\frac{\lambda_{t_k}}{\sigma^2(1-\lambda_{t_k})}\mathbf{I} + \hat{\mathcal{I}}_\pi\right)^{-1}$

output α_{t_k}, $\mathbf{a}_{t_k}$, $\mathbf{A}_{t_k}$

Algorithm 2 DPSMC (Interacting) with Normalising Constant Estimation

input base variance σ^2, initial aux. distribution q_0, particle count N_Y, sample count N_X, disc. step count K, LL step count N_{LL}, LL step size h_{LL}, ESS threshold κ, SDE forward kernel $\mathsf{K}_k^{\mathrm{sde}}$, and SDE backward kernel $\mathsf{L}_k^{\mathrm{sde}}$

for $i = 1, \ldots, N_X$ **do**
  $X_0^i \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$, $Y_0^{i,j} \sim q_0$ for all $j \in [N_Y]$
  $W_0^{i,j} = \tilde{\rho}_0(Y_0^{i,j}) / q_0(Y_0^{i,j})$ for all $j \in [N_Y]$
  $w_0^{i,j} = W_0^{i,j} / \sum_{l=1}^{N_Y} W_0^{i,l}$
  $S_0^i = \sum_{j=1}^{N_Y} w_0^{i,j} \varphi_{0,X_0^i}(Y_0^{i,j})$
  $M_0^i = \frac{1}{N_Y} \sum_{j=1}^{N_Y} W_0^{i,j}$, $M_{-1}^i := 1$ {Marginal density estimate}
  $\Lambda_0^i = M_0^i / \mathcal{N}(X_0^i; 0, \sigma^2 \mathbf{I})$ {X weight}
  $\tau_0^i = 0$ {Last time resampled}
end for
for $k = 1, \ldots, K-1$ **do**
  for $i = 1, \ldots, N_X$ **do**
    if $k < N_{\mathrm{LL}}$ **then**
      $X_k^i \sim \mathcal{N}(X_{k-1}^i + h_{\mathrm{LL}} S_{k-1}^i, 2h_{\mathrm{LL}} \mathbf{I})$ {LL}
    else
      $X_k^i \sim \mathsf{K}_k^{\mathrm{sde}}(\cdot \mid X_{k-1}^i, S_{k-1}^i, h)$ {SDE step}
    end if
    for $j = 1, \ldots, N_Y$ **do**
      $\bar{Y}_k^{i,j} \sim \mathsf{K}_k^{\mathrm{MALA}}(\cdot \mid Y_{k-1}^{i,j}, X_k^i)$ {MALA step, see Alg. 3}
      $W_k^{i,j} = W_{k-1}^{i,j} \frac{\mathcal{N}\left(\sqrt{\lambda_{t_k}} Y_{k-1}^{i,j}; X_k^i, \sigma^2(1-\lambda_{t_k})\right)}{\mathcal{N}\left(\sqrt{\lambda_{t_{k-1}}} Y_{k-1}^{i,j}; X_{k-1}^i, \sigma^2(1-\lambda_{t_{k-1}})\right)}$
    end for
  end for
  $w_k^{i,j} = W_k^{i,j} / \sum_{l=1}^{N_Y} W_k^{i,l}$ for all $i \in [N_X]$, $j \in [N_Y]$
  $\mathbf{A}_{t_k} = \mathrm{CVSE}(\{w_k^{i,j}\}, \{\bar{Y}_k^{i,j}\}, \{X_k^i\})$ {CV schedule estimation, see Alg. 4}
  for $i = 1, \ldots, N_X$ **do**
    $S_k^i = \sum_{j=1}^{N} w_k^{i,j} \phi_{t_k, X_k^i}^{\mathbf{A}_{t_k}}(\bar{Y}_k^{i,j})$ {Score estimation}
    $M_k^i = M_{\tau_{k-1}^i - 1}^i \frac{1}{N_Y} \sum_{j=1}^{N_Y} W_k^{i,j}$
    if $k < N_{\mathrm{LL}}$ **then**
      $\Lambda_k^i = \Lambda_{k-1}^i \frac{M_k^i \mathcal{N}(X_{k-1}^i; X_k^i + h_{\mathrm{LL}} S_k^i, 2h_{\mathrm{LL}} \mathbf{I})}{M_{k-1}^i \mathcal{N}(X_k^i; X_{k-1}^i + h_{\mathrm{LL}} S_{k-1}^i, 2h_{\mathrm{LL}} \mathbf{I})}$
    else
      $\Lambda_k^i = \Lambda_{k-1}^i \frac{M_k^i \mathsf{L}_{k-1}^{\mathrm{sde}}(X_{k-1}^i; X_k^i, h)}{M_{k-1}^i \mathsf{K}_k^{\mathrm{sde}}(X_k^i; X_{k-1}^i, S_{k-1}^i, h)}$
    end if
    if $\widehat{\mathrm{ESS}}(\{w_k^{i,j}\}_{i=1}^{N_Y}) < \kappa N_Y$ **then**
      $\{Y_k^{i,j}\}_{j=1}^{N_Y} \sim \sum_{l=1}^{N_Y} w_k^{i,l} \delta_{\bar{Y}_k^{i,l}}$ {Resampling}
      $W_k^{i,j} = 1$ for all $j \in [N_Y]$
      $\tau_k^i = k$
    else
      $Y_k^{i,j} = \bar{Y}_k^{i,j}$ for all $j \in [N_Y]$
      $\tau_k^i = \tau_{k-1}^i$
    end if
  end for
end for
for $i = 1, \ldots, N_X$ **do**
  $X_K^i \sim \mathsf{K}_K^{\mathrm{sde}}(\cdot \mid X_{K-1}^i, S_{K-1}^i, h)$ {SDE step}
end for
$\Lambda_K^i = \Lambda_{K-1}^i \frac{\exp(-V_\pi(X_K^i)) \mathsf{L}_{K-1}^{\mathrm{sde}}(X_{K-1}^i; X_K^i, h)}{M_{K-1}^i \mathsf{K}_K^{\mathrm{sde}}(X_K^i; X_{K-1}^i, S_{K-1}^i, h)}$
$\hat{Z}_\pi = \frac{1}{N_X} \sum_{i=1}^{N_X} \Lambda_K^i$

output approximate samples from the target $\{X_K^i\}_{i=1}^{N_X}$ and normalising constant estimate $\hat{Z}_\pi$

Taming terminal instabilities. While our scheme allows for single MALA steps per iteration, auxiliary variables have less time to equilibrate w.r.t. the next target. This can be detrimental when the sample makes large jumps at the end when the posterior contracts to a Dirac centred at the sample. In the absence of a finer discretisation, we turn to a simple heuristic that we find to work well empirically. It turns out that we consistently maintain our target acceptance ratio up to some tolerance, right until the posterior's Gaussian component dominates. We choose to heuristically halt the SMC sampler when the acceptance ratio goes below a threshold, which we set to 10%, and substitute score estimates with the target score itself. This only occurs in the final few discretisation steps and can be sufficiently avoided when using more than one MALA step.

F Theoretical Analysis

In this section, we provide proofs for Proposition 4 and Theorem 1 in the main text.

F.1 Score Estimation Error (Proof of Proposition 4)

As discussed in the main text, when $\lambda_{t_k} \approx 0$, the posterior is approximately the target, i.e. $\varrho_{t_k,X} \approx \pi$, which appears circular given we aim to draw samples from π. However, as observed by He et al. [32], when using DSI for score estimation, it is possible to rely on less accurate samples from $\varrho_{t,X}$ in the Wasserstein-2 sense for small t, while still keeping the overall MSE of the Monte Carlo estimator below a given threshold.

We analyse this behaviour for the weighted estimator constructed from the weighted particle system $\{(w_k^i, y_k^i)\}_{i=1}^N$ at the iteration k

$$S_k^N(x) = \mathbb{E}_{Y\sim\varrho_{t_k,x}^N}\left[\phi_{t_k,x}^{\alpha_{t_k}\mathbf{I}}(Y)\right] = \sum_{i=1}^N w_k^i \phi_{t_k,x}^{\alpha_{t_k}\mathbf{I}}(y_k^i),$$

where the test function

$$\phi_{t_k,x}^{\alpha_{t_k}\mathbf{I}}(y) = \alpha_{t_k}\frac{\sqrt{\lambda_{t_k}}\,y - x}{\sigma^2(1-\lambda_{t_k})} + (1-\alpha_{t_k})\frac{\nabla\log\pi(y)}{\sqrt{\lambda_{t_k}}},$$

is the control variate score expression.

We restate Proposition 4, which provides an upper bound on the L^2 error of the weighted score estimator (11), and present its proof.

Proposition 4 (Score estimation error)**.** *Suppose Assumption 1 holds and* $0 < \lambda_{t_k} < 1$*. Let* $\{X_k\}_{k=1}^K$ *be such that* $\mu_{t_k} = \mathrm{Law}(X_k)$ *for all* $k \geq 1$*. Given the score estimator* $S_k^N(X_k)$ *defined in* (11)*, assume that for some* $\delta(k,N) \geq 0$*,* $\mathbb{E}\left[W_2^2\left(\sum_{i=1}^N w_k^i \delta_{\bar{Y}_k^i}, \varrho_{t_k,X_k}\right)\right] \leq \delta(k,N)$ *holds where the expectation is over the joint law of* X_k*, the particles, and the weights. Then*

$$\mathbb{E}\left[\|\nabla\log\mu_{t_k}(X_k) - S_k^N(X_k)\|^2\right] \leq 2\,\delta(k,N)\left(\alpha_{t_k}^2\frac{\lambda_{t_k}}{\sigma^4(1-\lambda_{t_k})^2} + (1-\alpha_{t_k})^2\frac{L_\pi^2}{\lambda_{t_k}}\right). \tag{12}$$

Proof. Let $\mathcal{G}_k^N = \sigma(X_k, \{(w_k^i, \bar{Y}_k^i)\}_{i=1}^N)$ denote the sigma-field generated by X_k and the weighted particle system. Conditional on $\mathcal{G}_k^N$, the particle measure $\varrho_{t_k,X_k}^N = \sum_{i=1}^N w_k^i \delta_{\bar{Y}_k^i}$ is a deterministic probability measure, and so is ϱ_{t_k,X_k}, which is indexed by the realised value of X_k. Fix a realisation of this sigma-field: write $X_k = x$ and $\varrho_{t_k,x}^N = \sum_{i=1}^N w_k^i \delta_{y_k^i}$. Define the test functions

$$g_{k,x}(y) := \frac{\sqrt{\lambda_{t_k}}y - x}{\sigma^2(1-\lambda_{t_k})}, \quad h_k(y) := \frac{\nabla\log\pi(y)}{\sqrt{\lambda_{t_k}}},$$
$$f_{k,x}(y) := \alpha_{t_k} g_{k,x}(y) + (1-\alpha_{t_k})h_k(y) = \phi_{t_k,x}^{\alpha_{t_k}\mathbf{I}}(y).$$

By the score identity,

$$\nabla\log\mu_{t_k}(x) = \int f_{k,x}(u)\,\varrho_{t_k,x}(\mathrm{d}u), \quad S_k^N(x) = \int f_{k,x}(y)\,\varrho_{t_k,x}^N(\mathrm{d}y).$$

Let Γ be any coupling of $\varrho_{t_k,x}$ and $\varrho^N_{t_k,x}$, then

$$\left\|\nabla \log \mu_{t_k}(x) - S^N_k(x)\right\|^2 = \left\| \int \left(f_{k,x}(u) - f_{k,x}(y)\right) \Gamma(\mathrm{d}u, \mathrm{d}y) \right\|^2 \\ \leq \int \left\|f_{k,x}(u) - f_{k,x}(y)\right\|^2 \Gamma(\mathrm{d}u, \mathrm{d}y),$$

where we used Jensen's inequality. Next, we bound the Lipschitz constant of $f_{k,x}$. Since

$$\|g_{k,x}(u) - g_{k,x}(y)\| = \frac{\sqrt{\lambda_{t_k}}}{\sigma^2(1-\lambda_{t_k})}\|u-y\|,$$

and Assumption 1 implies

$$\|h_k(u) - h_k(y)\| \leq \frac{L_\pi}{\sqrt{\lambda_{t_k}}}\|u-y\|,$$

we have, using $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$,

$$\left\|f_{k,x}(u) - f_{k,x}(y)\right\|^2 \leq 2\left(\alpha^2_{t_k}\frac{\lambda_{t_k}}{\sigma^4(1-\lambda_{t_k})^2} + (1-\alpha_{t_k})^2\frac{L^2_\pi}{\lambda_{t_k}}\right)\|u-y\|^2.$$

Substituting this estimate into the previous display gives

$$\left\|\nabla \log \mu_{t_k}(x) - S^N_k(x)\right\|^2 \leq 2\left(\alpha^2_{t_k}\frac{\lambda_{t_k}}{\sigma^4(1-\lambda_{t_k})^2} + (1-\alpha_{t_k})^2\frac{L^2_\pi}{\lambda_{t_k}}\right)\int \|u-y\|^2 \Gamma(\mathrm{d}u, \mathrm{d}y).$$

Taking the infimum over all couplings Γ of $\varrho_{t_k,x}$ and $\varrho^N_{t_k,x}$ yields, for this fixed realisation,

$$\left\|\nabla \log \mu_{t_k}(x) - S^N_k(x)\right\|^2 \leq 2\left(\alpha^2_{t_k}\frac{\lambda_{t_k}}{\sigma^4(1-\lambda_{t_k})^2} + (1-\alpha_{t_k})^2\frac{L^2_\pi}{\lambda_{t_k}}\right) W_2^2\left(\varrho_{t_k,x}, \varrho^N_{t_k,x}\right).$$

Since the preceding inequality holds for every realisation of $\mathcal{G}^N_k$ for which the weights define a probability measure, it holds almost surely after substituting $x = X_k$ and $\varrho^N_{t_k,x} = \varrho^N_{t_k,X_k}$. Taking expectations with respect to the joint randomness of X_k, the particles, and the weights, and then using the assumed bound on the expected squared Wasserstein error gives

$$\mathbb{E}\left[\|\nabla \log \mu_{t_k}(X_k) - S^N_k(X_k)\|^2\right] \leq 2\,\delta(k,N)\left(\alpha^2_{t_k}\frac{\lambda_{t_k}}{\sigma^4(1-\lambda_{t_k})^2} + (1-\alpha_{t_k})^2\frac{L^2_\pi}{\lambda_{t_k}}\right).$$

□

Due to the choice of α^*_t, our estimator exhibits similar behaviour to that of He et al. [32] as $t \to 0$. In contrast, the setting considered in He et al. [32] (where $\alpha^*_t = 1$) requires an arbitrarily large number of samples as the algorithm approaches the target distribution ($t \to 1$) to maintain the L^2 bounded below some threshold. In our setting, however, as $t \to 1$, we can take $\alpha^*_t \approx 1 - \lambda_t$, implying that, in the last expression above, the term inside the parenthesis is of order $\mathcal{O}(1)$. This observation is consistent with the fact that DSI suffers from high variance in low-noise regimes.

The bound in Proposition 4 demonstrates that with sufficiently large N, we can ensure that

$$\sum_{k=0}^{K-1} h\,\mathbb{E}\left[\left\|\nabla \log \mu_{t_k}(X_k) - S^N_k(X_k)\right\|^2\right] \leq \varepsilon^2_{\text{score}}.$$

Therefore, we can apply different results in the literature to bound the final error of the algorithm.

F.2 Overall Algorithm Error (Proof of Theorem 1)

We restate Theorem 1 and present its proof.

Theorem 1. *Under Assumption 1 and further Assumption 2 for (ii) and (iii), the DPSMC algorithm with (i) OU time-reversal dynamics initialised at $X_{N_{LL}} \sim p_0$, (ii) SI dynamics initialised at $X_0 \sim \nu$, and (iii) DALD initialised at $X_0 \sim \nu$, satisfy respectively*

$$(i)\ \mathrm{KL}\left(\pi \parallel p_1\right) \lesssim \mathrm{KL}(\mu_0 \parallel p_0) + \tfrac{d}{K-N_{LL}}\left(1+T\right) + \varepsilon^2_{score},$$

$$(ii)\ \mathrm{KL}\left(\pi \parallel p_1\right) \lesssim \tfrac{dL^2}{K} + \varepsilon^2_{score},$$

$$(iii)\ \mathrm{KL}\left(\pi \parallel p_1\right) \lesssim \epsilon\left(1 + \tfrac{L^2}{K^2\epsilon^4}\right)(M_2 + d) + \tfrac{dL^2}{K\epsilon^2}\left(1 + \tfrac{L}{K\epsilon}\right) + \varepsilon^2_{score}.$$

Proof. Let p_1 denote the law of the algorithm at time $t = 1$ for different cases.

(i) Using the proof of Benton et al. [6, Theorem 1] in combination with Benton et al. [6, eq. (17)], we have that

$$\mathrm{KL}\left(\pi \parallel p_1\right) \leq \mathrm{KL}(\mu_0 \parallel p_0) + \varepsilon^2_{\text{score}} + \tfrac{d(1+T)}{K - N_{\text{LL}}}$$

While Benton et al. [6] assumes $T \geq 1$ to ensure the convergence error of the forward OU process is sufficiently small, our analysis does not require this since we explicitly account for the initialisation mismatch at $t = 0$ through $\mathrm{KL}(\mu_0 \parallel p_0)$. As the discretisation error of the reverse SDE is independent of the forward mixing time, the result holds for any $T > 0$.

(ii) Let $\mathbb{P}$ and $\mathbb{Q}$ denote the exact path measure of the continuous-time stochastic interpolant dynamics and the algorithmic path measure of its time-discretised counterpart, respectively. By the data processing inequality, the KL divergence between the final marginals is bounded by the divergence between their path measures

$$\mathrm{KL}(\pi \parallel p_1) \leq \mathrm{KL}(\mathbb{P} \parallel \mathbb{Q}).$$

Because $\mathbb{P}$ and $\mathbb{Q}$ share the same diffusion coefficient, an application of Girsanov's theorem bounds their divergence by the integrated expected squared difference of their drifts:

$$\mathrm{KL}(\mathbb{P} \parallel \mathbb{Q}) \lesssim \int \mathbb{E}_{\mathbb{Q}} \left[\left\| \nabla \log \mu_t(X_t) - S^N_k(X_{t_k}) \right\|^2 \right] dt.$$

Applying the triangle inequality, we decompose this drift error into a time discretisation error and a score estimation error

$$\begin{aligned} \mathbb{E}\left[\left\| \nabla \log \mu_t(X_t) - S^N_k(X_{t_k}) \right\|^2 \right] \leq & 2\mathbb{E}\left[\left\| \nabla \log \mu_t(X_t) - \nabla \log \mu_{t_k}(X_{t_k}) \right\|^2 \right] \\ & + 2\mathbb{E}\left[\left\| \nabla \log \mu_{t_k}(X_{t_k}) - S^N_k(X_{t_k}) \right\|^2 \right]. \end{aligned}$$

By definition, the integrated score estimation error over the K steps is bounded by ϵ^2_{score}. To bound the time discretisation error, we use the transition relationship $X_t = \alpha_t X_s + \sqrt{1 - \alpha_t^2} Z$ with $\alpha_t = \sqrt{\lambda_t / \lambda_s}$ for $t < s$. Under Assumptions 1 and 2, the exact score functions $\nabla \log \mu_t$ are L-Lipschitz [15, Lemma 3.2]. We can therefore apply Chen et al. [12, Lemma 16] across the K integration steps to bound the discretisation error

$$\int \mathbb{E}\left[\left\| \nabla \log \mu_t(X_t) - \nabla \log \mu_{t_k}(X_{t_k}) \right\|^2 \right] dt \lesssim \frac{dL^2}{K}.$$

Substituting both bounds into the Girsanov inequality yields the final result

$$\mathrm{KL}(\pi \parallel p_1) \leq \frac{dL^2}{K} + \epsilon^2_{score}.$$

(iii) This follows from Cordero-Encinar et al. [15, Theorem 3.4].

□

G Numerical Experiments

G.1 Methods and Hyperparameters

We detail here the hyperparameters of methods we examine. For a majority of them, we follow the settings and hyperparameter grids outlined in Grenioux et al. [29] when evaluating on targets that were previously considered. This includes incorporating information about the target, by way of R and τ, to each sampler. We recall their definition as coming from the following assumption.

Assumption 3 (Log concavity outside a compact)**.** *There exists $R > 0$ and $\tau > 0$ such that π is the convolution of μ and $\mathcal{N}(0, \tau^2 \mathbf{I})$, where μ is a distribution compactly supported on $B = \mathrm{B}(\mathbb{E}_\pi[X], R\sqrt{d})$, i.e. $\mu(\mathbb{R}^d \setminus B) = 0$.*

As our specific instantiation of GMM40 does not appear in Grenioux et al. [29] and may be different from other setups, we test every combination of hyperparameters in the defined grid of each sampler and take the best result. Unless otherwise specified, samplers with MALA kernels use an adaptive step size that is geometrically adjusted to maintain an acceptance ratio of 75%. Additionally, we set the following parameters fixed globally across all samplers (except RDMC) and benchmark targets:

- Number of discretisation steps: $K = 1024$.
- Maximum number of energy evaluations (per sample): $N_{\text{evals}} = 1.32 \times 10^5$.

AIS and SMC. Both AIS and SMC operate on the geometric path, i.e. $\rho_k(x) \propto \rho_0(x)^{1-\beta_k}\pi(x)^{\beta_k}$ for $k \in \{1, \dots, K\}$, where we have a Gaussian initial density $\rho_0(x) := \mathcal{N}(0, (R^2 d + \tau^2)\mathbf{I})$ and a linear annealing schedule $\beta_k = k/K$. Transitions between the path marginals are made via MALA transition kernels. Specifically, each sample runs 128 MALA steps per iteration. In SMC, the samples are adaptively resampled according to their importance weights whenever the ESS drops below half the number of particles. We find this performs better than the original settings in [29] where they use fewer MALA steps and resample at every time step.

RDMC. Here, the final time T in the (forward) OU process is treated as a hyperparameter. Following [34], we search for its values in the grid: $\{-\log(x) \mid x \in \{0.99, 0.95, 0.9, 0.8, 0.7\}\}$. We explain briefly how RDMC is implemented. To start, samples are drawn from $\mathcal{N}(0, (1-\exp(-2T))\mathbf{I})$ and 16 iterations of LL are performed. Here, and at every iteration afterwards, 128 samples are drawn from the Gaussian component of the posterior to form an IS estimate of the posterior mean. This serves as the initialisation of four chains that run 32 MALA steps, the first half of which are considered part of a warm-up and discarded. All combined, the 64 samples (auxiliary variables) are used to estimate the posterior mean and hence the score. Note that this totals to $(16+1024)\times(128+4\times32) = 2.66\times10^5$ energy evaluations. While this is more than the imposed budget, we stick to this formulation so we can choose the optimal hyperparameters from [29] for benchmarks other than GMM40. Its chosen settings across benchmarks are listed in Table 3.

Table 3: RDMC hyperparameters across the benchmark targets.

	GMM40	GMM40 ($d = 50$)	Rings	Funnel	Ionosphere	Sonar
Final OU Time (T)	-log(0.70)	-log(0.80)	-log(0.80)	-log(0.90)	-log(0.95)	-log(0.95)

SLIPS. We use the classic variant of SLIPS in our experiments. Similar to RDMC, it undergoes LL for the same number of iterations and estimates the posterior mean using four chains taking 32 MALA steps, half being warm-up steps. However, instead of using IS estimates to initialise the chains, it uses the positions of the chains in the previous time step. Furthermore, SLIPS starts at some later time t_0 for which both the path marginals and posterior distribution are assumed to be approximately log-concave. For GMM40, we perform a search along the grids $\eta = \{5.0, 5.7\}$ and $t_0 = \{0.1, 0.2, 0.4, 1.0, 1.2\}$, similar to the grid search suggested for mixtures in [29]. Its chosen settings across benchmarks are listed in Table 4.

Table 4: SLIPS hyperparameters across the benchmark targets.

	GMM40	GMM40 ($d = 50$)	Rings	Funnel	Ionosphere	Sonar
Log SNR Threshold (η)	5.0	5.7	4.6	5.0	5.0	5.0
Start Time (t_0)	1.20	0.10	1.20	1.00	0.03	0.03

DPSMC (OU). We set the Gaussian variance $\sigma^2 = R^2 d + \tau^2$, similar to AIS and SMC. Given we do not start at a Gaussian, we can search for a better time to initialise. We search for $\lambda_0 = \exp(-2T)$ uniformly in the grid $\{1/6, 2/6, \dots, 5/6\}$, discretising the unit interval. We believe this is more robust compared to the grid in RDMC, as we search along the path bridging the base Gaussian to the target rather than a heuristic range of values. The chosen values are listed in Table 5. Notably, we get similar values with RDMC.

DPSMC (DALD). Following our findings in Appendix D, we set the base Gaussian variance to be $\sigma^2 = \mathbb{E}_\pi[\|X\|^2]/d$ and the schedule to be $\lambda_t = \sin^2(\frac{\pi t}{2})$. DALD has ϵ as a hyperparameter. We

Table 5: DPSMC (OU) hyperparameters across the benchmark targets.

	GMM40	GMM40 ($d = 50$)	Rings	Funnel	Ionosphere	Sonar
Initial Schedule for LL (λ_0)	$3/6$	$5/6$	$3/6$	$3/6$	$5/6$	$5/6$
Final OU Time (T)	0.3465	0.0911	0.3465	0.3465	0.0911	0.0911

choose it as follows. Corollary A.1 indicates the following upper bound on the action

$$\mathcal{A}(\mu) \leq \frac{\pi^2}{4}\mathbb{E}_\pi[\|X\|^2],$$

which is achieved precisely under our choices of σ^2 and λ_t. Now, using this result in the workings of Cordero-Encinar et al. [15, Theorem 3.4], we can write

$$\mathrm{KL}(\mathbb{P} \parallel \mathbb{Q}) \lesssim \epsilon \left(1 + \frac{L^2}{K^2\epsilon^4}\right) \frac{\pi^2}{4}\mathbb{E}_\pi[\|X\|^2] + \frac{dL^2}{K\epsilon^2}\left(1 + \frac{L}{K\epsilon}\right) + \varepsilon^2_{\text{score}},$$

where $\mathbb{P}$ is the reference path measure, $\mathbb{Q}$ is the path measure of the continuous-interpolation of DALD, L is the supremum of the Lipschitz constants of $\nabla \log \mu_t$ along the path, and $\varepsilon^2_{\text{score}}$ is the score error. Loosely, we choose ϵ such that it minimises the first two terms. Taking the derivative with respect to ϵ and setting to zero, we have

$$\frac{\pi^2}{4}\left(1 - \frac{3L^2}{K^2\epsilon^4}\right)\mathbb{E}_\pi[\|X\|^2] - \frac{2dL^2}{K\epsilon^3} - \frac{3dL^3}{K^2\epsilon^4} = 0.$$

Assuming the $O(1/K^2)$ terms vanish, e.g. for large enough K, we then have

$$\frac{\pi^2}{4}\mathbb{E}_\pi[\|X\|^2] - \frac{2dL^2}{K\epsilon^3} = 0 \iff \frac{1}{\epsilon} = \left(\frac{K\mathbb{E}_\pi[\|X\|^2]\pi^2}{8dL^2}\right)^{1/3}.$$

We use this crude approximation and set $1/\epsilon = \xi\left(K\mathbb{E}_\pi[\|X\|^2]/d\right)^{1/3}$, for some hyperparameter ξ which we find in the grid

$$\left\{2^{\left[-2.5\cdot(1-\frac{x}{10})+3.5\cdot\frac{x}{10}\right]} \mid x \in \{0, 1, \ldots, 10\}\right\},$$

i.e. linearly interpolating in log-space. Notably, DALD is typically presented as running the dynamics

$$\mathrm{d}X_t = \nabla \log \mu_{t/T}(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}W_t, \quad X_0 \sim \nu,$$

for $t \in [0, T]$, where $T = 1/\epsilon$. For completeness, we present the chosen values for both quantities in Table 6. See Table 7 for the second moments of targets used to compute ϵ. For DALD with LL, we reuse the same ϵ but search for λ_0 as in DPSMC (OU). Because now we have $\lambda_0 \neq 0$, we recompute the schedule minimising the action upper bound with general boundary conditions to arrive at

$$\lambda_t = \frac{1}{2}\left(1 + \sin\left(\frac{\pi t}{2} + \arcsin(2\lambda_0 - 1)(1 - t)\right)\right).$$

Table 6: DPSMC (DALD) hyperparameters across the benchmark targets.

	GMM40	GMM40 ($d = 50$)	Rings	Funnel	Ionosphere	Sonar
ξ	$2^{3.5}$	$2^{2.9}$	$2^{-2.5}$	$2^{-0.1}$	$2^{-2.5}$	$2^{-2.5}$
Tracking Strength (ϵ)	0.0017	0.0025	0.3610	0.0652	0.5319	0.5435
Simulation Time (T)	584.25	387.66	2.77	15.33	1.88	1.84
Initial Schedule for LL (λ_0)	$4/6$	$4/6$	$4/6$	$4/6$	$1/6$	$1/6$

DPSMC (SI). Similar to DALD, we set the base Gaussian variance to be $\sigma^2 = \mathbb{E}_\pi[\|X\|^2]/d$ and the schedule to be $\lambda_t = \sin^2(\frac{\pi t}{2})$. Matching our OU implementation, we set the diffusion coefficient to be $\gamma_t = \sigma^2$. We choose to use an EM discretisation for the SDE in (2).

Table 7: The specific values of target information used to inform samplers across several target distributions.

Target	$\mathbb{E}_\pi[\|\|X\|\|^2]$	σ	R	τ
GMM40 ($d = 2$)	268.98	11.60	18.33	1
GMM40 ($d = 50$)	6840.25	11.70	13.41	1
Rings ($d = 2$)	7.52	1.94	$4/\sqrt{2}$	0.15
Funnel ($d = 10$)	43.34	2.08	$2.12/\sqrt{10}$	0
Ionosphere ($d = 35$)	≈ 41.25	1.09	$2.5/\sqrt{36}$	0
Sonar ($d = 61$)	≈ 67.25	1.05	$2.5/\sqrt{62}$	0

G.2 Target Distributions

We detail here the targets that were considered in our experiments. As we use the expected squared norm $\mathbb{E}_\pi[\|\|X\|\|^2]$ in our methods, we precisely write down their expressions or approximations if unavailable. A summary of their values are available in Table 7, including the choices for (R, τ) made by [29] to be used for the other samplers.

GMM40. We consider a uniform mixture of 40 Gaussians in d dimensions with identity covariance matrices and means sampled uniformly from a d-dimensional hypercube of side length 40. The density is exactly given by

$$\pi(x) = \frac{1}{40}\sum_{i=1}^{40}\mathcal{N}(x; \mathbf{m}_i, \mathbf{I}),$$

for $\mathbf{m}_i \sim U[-20, 20]^d$. We therefore have the expected squared norm equal to

$$\mathbb{E}_\pi[\|\|X\|\|^2] = d + \frac{1}{40}\sum_{i=1}^{40}\|\mathbf{m}_i\|^2.$$

We then appropriately set $R = \frac{1}{\sqrt{d}}\max_{i\in[40]}\{\|\mathbf{m}_i - \frac{1}{40}\sum_{j=1}^{40}\mathbf{m}_j\|\}$ and $\tau = 1$. In our experiments, we fix the seed (0) when sampling the mean vectors. Under these choices, we arrive at values shown in Table 7.

Rings. The rings distribution [29] is a radially symmetric distribution featuring four concentric rings of different radii as the regions with high density. First, we define two univariate distributions: a radial one p_r given by

$$p_r(x) = \frac{1}{4}\sum_{i=1}^{4}\mathcal{N}(x; i, 0.15^2),$$

and an angular one p_θ that is a uniform distribution over $[0, 2\pi]$. The rings distribution is precisely the inverse polar reparameterisation of the joint distribution of both radial and angular components

$$\pi(x) = p_r(\|x\|)p_\theta(\tan^{-1}(x_1/x_2)).$$

Given the rotational symmetry, it suffices to look at the radial component to compute the expected squared norm. We have

$$\mathbb{E}_\pi[\|\|X\|\|^2] = \mathbb{E}_{p_r}[X^2] = \frac{1}{4}\sum_{i=1}^{4}(i^2 + 0.15^2) = 7.5225.$$

Funnel. The funnel distribution [40] is a hierarchical distribution featuring an exponentially narrowing region of high density. It is exactly given by

$$\pi(x) = \mathcal{N}(x_1; 0, \eta^2)\prod_{i=2}^{10}\mathcal{N}(x_i; 0, \exp(x_1)),$$

for some η^2. While Grenioux et al. [29] claim to use $\eta = 3$, we found their code to have actually used $\eta^2 = 3$ when assessing their sampler and its hyperparameters. We accordingly use $\eta^2 = 3$ for our experiments. We now solve for its expected squared norm. We have that

$$\begin{aligned}\mathbb{E}_\pi[\|X\|^2] &= \eta^2 + \int \tfrac{1}{\sqrt{2\pi\eta^2}} \exp\left(-\tfrac{x_1^2}{2\eta^2}\right) \int \tfrac{1}{\sqrt{2\pi \exp(x_1)^9}} \exp\left(-\tfrac{\|x_{2:10}\|^2}{2\exp(x_1)}\right) \|x_{2:10}\|^2 \mathrm{d}x_{2:10}\mathrm{d}x_1 \\ &= \eta^2 + \int \tfrac{1}{\sqrt{2\pi\eta^2}} \exp\left(-\tfrac{x_1^2}{2\eta^2}\right) (9\exp(x_1))\,\mathrm{d}x_1 \\ &= \eta^2 + 9\exp(\eta^2/2) \approx 43.34\end{aligned}$$

where we used the fact that the MGF for a univariate gaussian $p(x) = \mathcal{N}(x; \mu, \sigma^2)$ is $\mathbb{E}_p[\exp(tX)] = \exp(\mu t + t^2\sigma^2/2)$.

Bayesian logistic regression. Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^M$ with vector-valued features $x_i \in \mathbb{R}^d$ and binary outcomes $y_i \in \{0, 1\}$ for $i \in [M]$, a Bayesian logistic regression model is given by a prior on its weight vector $w \in \mathbb{R}^d$ and intercept $b \in \mathbb{R}$ which we choose to be

$$p(w, b) = \mathcal{N}(w; 0, \mathbf{I})\mathcal{N}(b; 0, 2.5^2),$$

and a likelihood for a pair (x, y) in terms of the model parameters

$$p(y \mid x; w, b) = \text{Bernoulli}\left(y; \sigma(w^\top x + b)\right),$$

where $\sigma(x) = \frac{\exp(x)}{1+\exp(x)}$ is the sigmoid function. The goal is to sample from the posterior distribution

$$\pi(w, b) := p(w, b \mid \mathcal{D}) \propto p(\mathcal{D} \mid w, b)p(w, b) = \left[\prod_{i=1}^M p(y_i \mid x_i; w, b)\right] p(w, b).$$

Instead of computing the expected squared norm of the model parameters with respect to the posterior, we compute our statistics based on the prior directly, i.e. we have

$$\mathbb{E}_p[\|W\|^2 + B^2] = d \cdot 1 + 2.5^2 = d + 6.25.$$

G.3 Normalising Constant Estimation

We briefly detail our setup for estimating normalising constants. The necessary modifications to our method are highlighted in Algorithm 2. Here, we have included an additional backwards kernel $\mathsf{L}_k^{\text{sde}}$ for defining the extended target space. In the OU case, we choose EM discretisations of the forward OU SDE and its time-reversal, i.e.

$$\begin{aligned}\mathsf{K}_{k+1}^{\text{sde}}(X_{k+1} \mid X_k, S_k, h) &= \mathcal{N}(X_{k+1}; X_k + hT(X_k + 2\sigma^2 S_k), 2hT\sigma^2\mathbf{I})), \\ \mathsf{L}_k^{\text{sde}}(X_k \mid X_{k+1}, h) &= \mathcal{N}(X_k; X_{k+1} - hTX_{k+1}, 2hT\sigma^2\mathbf{I}),\end{aligned}$$

which we found to be more stable than their EI counterparts.

In this experiment, we follow roughly the same number of energy evaluations and discretisation as RDSMC [59]. For Rings, we run AIS for 32 steps between $\mathcal{N}(x; 0, \tilde{\sigma}^2 \min(1, (1-\lambda_0)/\lambda_0)\mathbf{I})$ and ϱ_{0,X_0} to initialise our auxiliary variables, then we run our algorithm for a smaller number of discretisation steps, $K = 124$, but compensate with 8 MALA steps per iteration. This gives the same number of energy evaluations as specified in our previous benchmarks. For Funnel, we run 32×5 AIS steps and 8×5 MALA steps per iteration. We choose to do an AIS initialisation as it helps provide an accurate estimate for the initial normalising constant. Otherwise, simple IS may provide a high variance estimate that drowns the contribution of the remaining telescoping ratio. Further, unlike RDSMC, we perform an LL initialisation with $N_{\text{LL}} = 8$ steps, with each step being accounted for in the weights using symmetric forward and backward ULA kernels.

As noticed by Wu et al. [59], the marginal density estimates may be unstable at certain times, especially when the posterior coverage is poor, as is typically the case in high dimensions. While RDSMC has a hyperparameter for when resampling is allowed to take place, we choose to not resample for simplicity. We previously mentioned taming terminal instabilities by halting the SMC sampler when the acceptance ratio drops below a threshold and using the true target scores. In this setting where we still need marginal density estimates, we perform SNIS with the Gaussian

component of the posterior as the proposal. This performs quite well, given it has a dominant contribution for λ_t close to 1.

We perform a hyperparameter search for λ_0 in the same uniform grid $\{1/6, 2/6, \ldots, 5/6\}$. The selected values are found in Table 8.

Table 8: DPSMC (OU) hyperparameters for normalising constant estimation.

	Rings	Funnel
Initial Schedule for LL (λ_0)	$4/6$	$4/6$
Final OU Time ($T = -\log(\lambda_0)/2$)	0.2027	0.2027

H Ablations and Additional Experiments

H.1 Choice of Score Identity

In the main text, we proposed control variate schedules that were independent of position, yielding a score identity we term MCVSI. Our experiments made use of this identity with the matrix schedule. Here, we investigate how using the other score identities affects the performance of our algorithm.

H.1.1 Benchmarking Other Score Identities

We run our benchmarks on the other set of identities: DSI, TSI, MSI [18, 31], the control-variate score identity (CVSI) [35, 37] (as written in Lemma A.3), (scalar) MCVSI, and (diagonal) DMCVSI. We test the identities on two regimes: one with many auxiliary variables (same setup as our main experiments) and one with fewer. This is to see whether score identities that rely on estimation do worse with fewer auxiliary variables than those that do not, e.g. MSI, and whether MCVSI, which uses the same schedule for each sample, loses some fidelity in contrast to CVSI, which estimates schedules on a per-sample basis. Table 9 summarises our findings.

Overall, we find that: (i) our matrix schedule was best in handling the anisotropic Funnel target with the diagonal schedule in second, (ii) DSI and TSI alone are quite unstable (without clipping score estimates), (iii) identities relying on estimation work even with few auxiliary vriables, and (iv) MCVSI provides comparable if not better performance than CVSI.

Table 9: Performance of DPSMC (DALD) using different settings and control variate schedules across ten seeds. Best result under each DPSMC setting is in **bold**, and second-best is underlined.

Setting / Score Identity	GMM40 ($d = 2$) W_2 (↓)	GMM40 ($d = 50$) W_2 (↓)	Rings ($d = 2$) W_2 (↓)	Funnel ($d = 10$) KS (↓)	Ionosphere ($d = 35$) Log-LL (↑)	Sonar ($d = 61$) Log-LL (↑)
$K = 1024, N_Y = 128$						
DSI	5.58±1.72	29.30±1.47	**0.19±0.02**	0.094±0.004	-84.81±1.23	-108.58±0.09
TSI	**1.82±0.26**	29.08±1.07	**0.19±0.02**	0.097±0.003	-438.94±2.75	-330.93±0.89
MSI	1.89±0.26	29.59±1.00	0.21±0.02	0.069±0.003	-83.96±0.11	-109.06±0.07
CVSI	1.83±0.27	29.62±0.76	0.21±0.02	0.073±0.006	-83.76±0.10	-107.92±0.07
MCVSI	1.89±0.28	**29.05±1.51**	0.21±0.02	0.070±0.003	**-83.71±0.10**	**-107.67±0.07**
DMCVSI	1.94±0.25	29.40±1.26	0.21±0.02	0.063±0.004	-83.74±0.10	-108.01±0.05
MMCVSI	1.92±0.22	**29.05±1.51**	0.20±0.02	**0.062±0.003**	-83.92±0.09	-108.84±0.09
$K = 8192, N_Y = 16$						
DSI	2.06±3.32	30.04±1.82	nan	0.154±0.022	-91.22±1.08	-110.31±0.15
TSI	**1.42±0.34**	nan	nan	0.189±0.004	-98.21±0.20	-119.84±0.19
MSI	1.43±0.25	30.39±1.35	0.22±0.02	0.041±0.002	-86.00±0.09	-109.47±0.11
CVSI	1.44±0.27	**29.47±0.89**	0.20±0.02	0.045±0.003	-86.02±0.09	-109.39±0.11
MCVSI	1.44±0.23	30.20±1.31	**0.18±0.22**	0.036±0.005	**-85.92±0.09**	**-109.22±0.11**
DMCVSI	1.47±0.29	30.20±1.31	0.20±0.02	0.035±0.005	-85.93±0.10	-109.29±0.07
MMCVSI	1.46±0.26	29.94±1.19	**0.18±0.01**	**0.031±0.002**	-85.96±0.09	-109.43±0.10

H.1.2 High-Dimensional Scaling of CV Schedules

We provide three varying levels of our control variate schedules in Appendix C.2. While choosing the matrix schedule is provably best in the idealised setting, it is unclear in practice when to choose between scalar, diagonal, or matrix schedules because of differences in their dimension, estimation fidelity, numerical instabilities, and complexity. We consider two targets with different score covariance structures: Funnel [40], which is diagonal, and anisotropic Gaussians with non-zero off-diagonal entries. We probe DPSMC (SI) for this experiment, under increasing dimension, featuring the same settings and budget as in our benchmarks, regardless of dimension.

Funnel. In Table 10, we report the sliced KS distance, mean ESS, and runtime, averaged over three seeds. We observe two things at very high dimensions. First, the scalar schedule becomes unstable near the end of the path ($\lambda_t \to 1$) and diverges. Note that Funnel's score covariance is $\mathcal{I}_\pi = \mathrm{diag}(\frac{1}{\eta} + \frac{d}{2}, e^{\eta/2}, \ldots, e^{\eta/2})$. The score variance of the first coordinate grows with d, yet the scalar schedule has no per-coordinate degree of freedom to absorb it. Under DPSMC (OU) we found the scalar schedule remains stable, suggesting it is SI's transport, not the control variate alone, that tips it into divergence. Second, the diagonal schedule slightly outperforms the matrix one, presumably because the matrix schedule adds variance from off-diagonal terms that are identically zero.

Table 10: Dimension scaling of DPSMC (SI) on Funnel using different control variate schedules across three seeds. The KS floor (sliced KS between two independent sets of 4096 exact draws) is $\approx$0.016 for all d. Best KS under each dimension is in **bold**, and second-best is underlined.

	Scalar			Diagonal			Matrix		
d	KS (↓)	ESS (↑)	Runtime (s) (↓)	KS (↓)	ESS (↑)	Runtime (s) (↓)	KS (↓)	ESS (↑)	Runtime (s) (↓)
10	**0.022**$\pm$**0.001**	91.6$\pm$0.0	57$\pm$1	0.023$\pm$0.001	91.6$\pm$0.0	52$\pm$1	0.023$\pm$0.001	91.6$\pm$0.0	56$\pm$0
20	**0.036**$\pm$**0.001**	86.9$\pm$0.0	57$\pm$1	0.037$\pm$0.002	86.8$\pm$0.0	55$\pm$2	0.037$\pm$0.002	86.8$\pm$0.0	52$\pm$2
40	0.074$\pm$0.002	83.0$\pm$0.0	67$\pm$2	0.070$\pm$0.002	82.6$\pm$0.3	70$\pm$1	**0.069**$\pm$**0.002**	82.8$\pm$0.3	66$\pm$4
80	0.110$\pm$0.002	79.7$\pm$0.0	86$\pm$2	0.090$\pm$0.002	79.7$\pm$0.0	84$\pm$3	**0.089**$\pm$**0.003**	79.8$\pm$0.0	86$\pm$1
160	0.125$\pm$0.002	77.7$\pm$0.0	126$\pm$1	0.099$\pm$0.002	77.3$\pm$0.1	126$\pm$2	**0.098**$\pm$**0.003**	77.3$\pm$0.1	145$\pm$1
320	nan	–	–	**0.107**$\pm$**0.002**	75.9$\pm$0.1	205$\pm$3	0.108$\pm$0.002	75.8$\pm$0.0	216$\pm$1
640	nan	–	–	**0.109**$\pm$**0.001**	75.3$\pm$0.0	324$\pm$7	0.115$\pm$0.002	75.3$\pm$0.0	361$\pm$2

Anisotropic Gaussian. We consider a zero-mean Gaussian $\mathcal{N}(0, \Sigma)$ with covariance $\Sigma = Q\Lambda Q^\top$, for a rotation matrix Q fixed per dimension, and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$, for $\lambda_i = \kappa^{\frac{i-1}{d-1} - \frac{1}{2}}$ equally log-spaced, fixing the condition number $\kappa = 10$. In Table 11, we report relative covariance error $\mathrm{RCE}(\Sigma, \hat{\Sigma}) = \|\Sigma - \hat{\Sigma}\|_F / \|\Sigma\|_F$, average relative score error along the path, and mean ESS, averaged over three seeds. We omit the runtimes which scale similarly as in Table 10. We find all schedules reach the RCE floor, with the matrix schedule having the smallest relative score error.

Table 11: Dimension scaling of DPSMC (SI) on an anisotropic Gaussian with condition number 10 using different control variate schedules across three seeds. We report relative covariance error (RCE) and average relative score error along the path (RSE). The RCE floor is the RCE of exact samples. Best RSE under each dimension is in **bold**, and second-best is underlined.

		Scalar		Diagonal		Matrix	
d	RCE floor	RCE (↓)	RSE (↓)	RCE (↓)	RSE (↓)	RCE (↓)	RSE (↓)
10	0.043	0.042$\pm$0.004	2.75e-2$\pm$1.0e-4	0.042$\pm$0.004	2.53e-2$\pm$2.1e-5	0.041$\pm$0.004	**2.42e-4**$\pm$**2.2e-5**
20	0.061	0.055$\pm$0.003	2.58e-2$\pm$2.0e-4	0.055$\pm$0.003	2.49e-2$\pm$1.9e-4	0.055$\pm$0.003	**3.42e-4**$\pm$**3.7e-6**
40	0.087	0.086$\pm$0.002	2.54e-2$\pm$1.2e-4	0.086$\pm$0.003	2.46e-2$\pm$1.2e-4	0.086$\pm$0.003	**4.94e-4**$\pm$**1.6e-5**
80	0.118	0.117$\pm$0.002	2.52e-2$\pm$3.6e-5	0.117$\pm$0.002	2.47e-2$\pm$8.9e-5	0.117$\pm$0.002	**7.06e-4**$\pm$**5.0e-6**
160	0.167	0.166$\pm$0.002	2.50e-2$\pm$5.6e-5	0.166$\pm$0.003	2.49e-2$\pm$7.8e-5	0.166$\pm$0.003	**1.00e-3**$\pm$**6.5e-6**
320	0.236	0.236$\pm$0.001	2.49e-2$\pm$2.6e-5	0.236$\pm$0.001	2.49e-2$\pm$3.6e-5	0.237$\pm$0.000	**1.41e-3**$\pm$**3.8e-6**
640	0.333	0.331$\pm$0.001	2.49e-2$\pm$4.8e-6	0.331$\pm$0.001	2.49e-2$\pm$3.0e-5	0.334$\pm$0.001	**2.00e-3**$\pm$**6.0e-6**

Informed by our findings, a rule of thumb is to choose between the three depending on the perceived structure of the target score covariance. As we have seen, the scalar schedule can diverge when there

is inherent anisotropy not being captured. On the more practical side, the matrix schedule requires keeping in memory the entire covariance matrix, which becomes expensive as dimension is increased, not to mention the matrix inversion involved. The diagonal schedule hence strikes a balance between fidelity and ease in computation, making it a solid first choice.

H.2 Wall-clock Times

We measure wall-clock times for running SMC, SLIPS, and DPSMC (DALD) across all the benchmarks on a single Nvidia RTX 6000 Ada GPU. Each result is in seconds and reported as an average of five seeds, with each seed producing 4096 samples. Note that all implementations are in PyTorch. As in Table 12, we find empirically that DPSMC is significantly faster than other samplers on the toy targets and comparable in logistic regression. In principle, one can choose different hyperparameters for the competing samplers to match the speed of DPSMC. Doing so for SMC and SLIPS typically results in worse performance compared to those reported and DPSMC.

Table 12: Wall-clock times of samplers in seconds averaged across five seeds. Best result is in **bold**.

Algorithm	Batched Evals	GMM40 ($d = 2$)	GMM40 ($d = 50$)	Rings ($d = 2$)	Funnel ($d = 10$)	Ionosphere ($d = 35$)	Sonar ($d = 61$)
SMC	131K	$321.68_{\pm 3.07}$	$364.44_{\pm 21.00}$	$603.88_{\pm 16.27}$	$363.07_{\pm 5.96}$	$357.12_{\pm 27.16}$	$400.10_{\pm 13.68}$
SLIPS	32K	$77.25_{\pm 2.43}$	$100.68_{\pm 8.20}$	$170.59_{\pm 11.72}$	$150.56_{\pm 10.65}$	$89.12_{\pm 1.43}$	$\mathbf{94.70_{\pm 2.00}}$
DPSMC	**1K**	$\mathbf{9.66_{\pm 1.99}}$	$\mathbf{53.77_{\pm 1.27}}$	$\mathbf{9.51_{\pm 0.29}}$	$\mathbf{10.60_{\pm 0.48}}$	$\mathbf{83.14_{\pm 1.24}}$	$99.77_{\pm 0.71}$

H.3 Comparison with Parallel Tempering

Parallel tempering (PT) [50, 27] has largely remained as the gold standard amongst sampling methods. It defines a sequence of inverse temperatures $0 < \beta_1 < \ldots < \beta_{N_B} = 1$ and subsequently targets the joint distribution $p(x_1, \ldots, x_{N_B}) \propto \prod_{i=1}^{N_B} \pi(x_i)^{\beta_i}$. To do this, it runs several Markov chains in parallel, one at each temperature level, often via Hamiltonian Monte Carlo (HMC) [41] samplers. After every step, chains at different temperature levels, e.g. adjacent ones, are swapped according to some probability that leaves the joint distribution invariant. This scheme allows the chain at the normal temperature to traverse between modes through swaps with chains at higher temperatures where the energy barriers present are significantly lower.

Now, we describe our choices for its settings. We fix the number of chains to be $N_B = 16$ and set the inverse temperatures to be $\beta_i = 10^{-(i-1)/5}$. While we would normally run a single instance of PT and take the positions of the coldest chain at every other step, we chose to run 16 parallel instances to make runtimes tractable. Each chain runs an HMC sampler. Simulating Hamiltonian dynamics requires specifying the mass matrix M, the step size h, and the number of leapfrog steps L. We set the mass matrix to be the identity $M = \mathbf{I}$ and choose a step size h that achieves roughly 0.65 acceptance ratio [41]. First, we initialise 64 chains per temperature level from $\mathcal{N}(0, (R^2 d + \tau^2)\mathbf{I})$ and do 200 warm-up steps to geometrically adjust the step size (per temperature) to achieve the target acceptance ratio. Then, we only keep 16 samples (number of instances) at each temperature level, and, for every subsequent HMC step, we use the same step size obtained after the warm-up. We choose the number of leapfrog steps L to be equal to 128, and we take the coldest chain's position after every 64 HMC steps and treat them as our samples. To be precise, the total energy evaluations (which should be less than $N_{\text{evals}} \times 4096$) is the product of the following quantities: number of instances (16), number of chains/temperatures (16), number of HMC steps (64), number of leapfrog steps (128), number of samples per instance (256).

We include our findings in Table 1 for completeness but remark that PT takes significantly longer to complete, despite having parallel instances in the same temperatures, each using a fraction of the total energy budget.

H.4 Initialisation Dependency

We discuss our setup for measuring the initialisation dependency of our algorithm, as documented in Figure 3 (left) in the main text. Recall that $\varrho_{t,x}$ is the density of X_1 given $X_t = x$. It is then reasonable to expect that poor coverage of posterior modes by initial auxiliary variables can lead to

final X_1 samples missing the same set of modes. We aim to understand how our sampler behaves precisely under this poor initialisation regime. To quantify mode coverage, we first describe a means for estimating the mode weights of a mixture distribution.

Mode weight estimation. Let $\pi(x) = \sum_{i=1}^{N_m} w_i \pi_i(x)$ be some arbitrary mixture distribution, with mixture weights w_i and component distributions π_i. Given the final samples $\{X_K^j\}_{j=1}^N$ of our algorithm, we say the predicted ith mode weight is given by

$$\hat{w}_i = \frac{1}{N} \sum_{j=1}^{N} \frac{w_i \pi_i(X_K^j)}{\sum_{k=1}^{N_m} w_k \pi_k(X_K^j)}. \tag{20}$$

Later, we either measure individual weights or the mode TVD, i.e. $\mathrm{MTVD}(w, \hat{w}) = \frac{1}{2}\|w - \hat{w}\|_1$.

Poor initialisation recovery. Inspired by Grenioux et al. [29], we consider the five-dimensional uneven bi-modal Gaussian mixture

$$\pi(x) = \tfrac{9}{10}\mathcal{N}(x; -\tfrac{2a}{3}\mathbf{1}_5, \tfrac{1}{10}\mathbf{I}_5) + \tfrac{1}{10}\mathcal{N}(x; \tfrac{4a}{3}\mathbf{1}_5, \tfrac{1}{10}\mathbf{I}_5). \tag{21}$$

We run DPSMC (DALD) and DPSMC (SI), which start at $\lambda_0 = 0$, with their auxiliary variables initialised at the light mode, i.e. $q_0(x) = \mathcal{N}(x; \frac{4a}{3}\mathbf{1}_5, \frac{1}{10}\mathbf{I}_5)$, and measure how the final samples capture the heavy mode weight $w_1 = 0.9$. Here, a (mode separation) is increased so the ratio $r = R^2 d/\tau^2$, with $R = 1.8a$ and $\tau^2 = 0.1$, exceeds the known threshold for the window's existence, i.e. $r < 6.48$ [29, Proposition 19]. To provide a concrete baseline, we also run a single MALA chain, initialised at the light mode, with a comparable energy budget. To make the MALA chain comparison feasible, however, we need to downsize our sampler to settings $N_X = 256, N_Y = 16, K = 1024$, i.e. MALA runs for $N_X \times N_Y \times K = 4,194,304$ steps. For simplicity, we run DALD fixing $\xi = 1.0$. While this may be suboptimal, the goal of this experiment is to see how DALD and SI differ in their recovery trends. We can, in fact, significantly improve mode coverage with sufficient tuning. In Table 13, we report the mode weight estimates for DALD and SI under their original budget, in line with Figure 3 (left), which features similar results to the downsized budget. Despite an adversarial initialisation, we find DALD is able to recover and reproduce the heavy mode weight up to some large separation a, where particles eventually become stuck in their initial mode. SI recovers to some extent but largely maintains its error. MALA also recovers the heavy mode weight for moderately separated mixtures. At $r = 64.0$, however, it mixes poorly despite the metric suggesting otherwise. Given it has significantly more steps, it escapes the light mode but remains stuck at the heavy mode for the rest of the simulation. DALD also has particles that reach the heavy mode, but it behaves fundamentally different from a single Markov chain where this greatly influences the entire trajectory or ensemble of samples.

Table 13: Estimated heavy mode weight $\hat{w}_1$ (true value 0.9) on the bi-modal mixture in (21), with an adversarial initialisation, across ten seeds. The ratio $r = R^2 d/\tau^2$ grows with the mode separation a. * Here, MALA mixes poorly and never leaves the heavy mode, resulting in an overestimation.

Method / r	0.5	1.0	2.0	4.0	8.0
DPSMC (DALD)	$0.900_{\pm 0.001}$	$0.900_{\pm 0.002}$	$0.897_{\pm 0.004}$	$0.898_{\pm 0.004}$	$0.899_{\pm 0.007}$
DPSMC (SI)	$0.881_{\pm 0.002}$	$0.858_{\pm 0.004}$	$0.806_{\pm 0.007}$	$0.678_{\pm 0.011}$	$0.416_{\pm 0.017}$
MALA	$0.900_{\pm 0.000}$	$0.900_{\pm 0.000}$	$0.900_{\pm 0.000}$	$0.900_{\pm 0.000}$	$0.900_{\pm 0.000}$
Method / r	16.0	32.0	64.0	128.0	256.0
DPSMC (DALD)	$0.900_{\pm 0.001}$	$0.879_{\pm 0.008}$	$0.024_{\pm 0.004}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$
DPSMC (SI)	$0.100_{\pm 0.008}$	$0.001_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$
MALA	$0.901_{\pm 0.001}$	$0.897_{\pm 0.003}$	$0.921_{\pm 0.069}$*	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$

H.5 Posterior and Score Error

We discuss our setup for measuring the empirical convergence rates of the posterior and score error, as shown in Figure 3 (middle and right) in the main text. Here, our goal is to understand how the score MSE and W_2^2 distance between the SMC empirical distribution and posterior are controlled by the particle count. To measure this empirically, we run DPSMC (DALD) under the standard settings on the same bi-modal mixture target in (21), under an increasing particle count and separation, and

average over the samples and auxiliary variables at $\lambda_k \approx 0.5$, to form Monte Carlo estimates for the quantities: $\mathbb{E}[W_2^2(\sum_i w_k^i \delta_{\bar{Y}_k^i}, \varrho_{t_k,X_k})]$ and $\mathbb{E}[\|\nabla \log \mu_{t_k}(X_k) - S_k^N(X_k)\|^2]$. Table 14 shows our findings, including the slope from an OLS fit of the log estimate on $\log N$. As mentioned in the main text, results for the modest separation regime closely recapitulate existing rates. Notably, for the high separation regime, $r = 64.0$, we find the score MSE has a rather flat trend whereas the W_2^2 from the posterior maintains a sizeable slope. In Proposition 4, we bound the score MSE above by the W_2^2 from the posterior multiplied by some constant. We believe the difference in rates can be explained by the constant depending on the target, i.e. through the Lipschitz constant and optimal control variate coefficients used in the simulation, which is changing as the separation is increased. The score MSE standard deviation is also fairly substantial here. We remark that our assumption in Proposition 4 is uniform in time whereas we conduct this experiment for a single time point.

Table 14: Monte Carlo estimates of the two quantities in Proposition 4 on the uneven bi-modal mixture in (21), averaged over three runs of DALD. The ratio $r = R^2 d/\tau^2$ grows with the mode separation a. The slope comes from an OLS fit of the log estimate on $\log N$.

	W_2^2 from Posterior		
N / r	1.0	16.0	64.0
16	2.38e-1$_{\pm 1.9e\text{-}5}$	2.83e-1$_{\pm 3.0e\text{-}4}$	3.76e-1$_{\pm 2.8e\text{-}3}$
64	1.40e-1$_{\pm 3.4e\text{-}4}$	1.65e-1$_{\pm 4.1e\text{-}4}$	2.45e-1$_{\pm 9.5e\text{-}4}$
256	8.12e-2$_{\pm 2.0e\text{-}4}$	9.52e-2$_{\pm 1.0e\text{-}4}$	1.67e-1$_{\pm 8.0e\text{-}3}$
1024	4.70e-2$_{\pm 2.7e\text{-}5}$	5.50e-2$_{\pm 1.7e\text{-}4}$	1.08e-1$_{\pm 1.9e\text{-}3}$
Slope	-0.390	-0.394	-0.298

	Score MSE		
N / r	1.0	16.0	64.0
16	1.71e-3$_{\pm 5.4e\text{-}5}$	4.89e-1$_{\pm 4.5e\text{-}2}$	1.31e+0$_{\pm 1.1e\text{-}1}$
64	5.21e-4$_{\pm 9.2e\text{-}6}$	2.26e-1$_{\pm 4.9e\text{-}3}$	1.40e+0$_{\pm 2.9e\text{-}2}$
256	1.49e-4$_{\pm 2.0e\text{-}6}$	1.01e-1$_{\pm 8.1e\text{-}3}$	1.42e+0$_{\pm 1.9e\text{-}1}$
1024	3.96e-5$_{\pm 2.2e\text{-}6}$	4.35e-2$_{\pm 4.0e\text{-}3}$	1.08e+0$_{\pm 6.7e\text{-}2}$
Slope	-0.906	-0.582	-0.041

H.6 Hyperparameter Sensitivity

Our algorithms require various hyperparameters whose choices may be non-trivial but greatly influence performance. Here, we conduct a simple experiment to better understand how such choices can be informed. We consider running DPSMC (OU) through different settings for the LL initialisation and the underlying SMC sampler on GMM40 and Funnel targets. For DALD, we point to our discussion in Appendix G.1 which selects a value for ϵ that heuristically minimises the overall algorithm bound. We measure performance under GMM40 in MTVD as described in Appendix H.4.

LL initialisation. Here, we test sensitivity to the LL step count N_{LL} and the initial time λ_0 which are responsible for LL initialisation. We fix the energy budget and hence have N_{LL} steps for LL and $K - N_{\text{LL}}$ steps for the remainder of the path. In Table 15, we find target-dependent sweet-spots for initialisation, with results most sensitive to λ_0. Both parameters trade off between better-initialised X samples (small λ_0 and large N_{LL}) and finer discretisation of the remaining path (large λ_0 and small N_{LL}). The auxiliary variable ESS is highest when the initial marginal is close to the base distribution and no LL is performed. In general, there are values l_μ and l_ρ such that μ_t is log-concave when $\lambda_t < l_\mu$ and $\varrho_{t,x}$ is log-concave when $\lambda_t > l_\rho$. When the duality of log-concavity [29] holds, i.e. $l_\rho < l_\mu$, we want to initialise $\lambda_0 \in (l_\rho, l_\mu)$. However, the window's existence and location are target dependent and generally unavailable unless we know, e.g., the Lipschitz constant of the target score (for l_ρ) or R and τ [29] (for l_ρ and l_μ bounds). When doing LL, we expect N_{LL} to depend on how fast the chains initialised at ν can equilibrate to μ_{λ_0}.

SMC sampler. Here, we test sensitivity to the particle count N_Y, MALA step count N_{MALA}, and ESS threshold κ which encompass the SMC sampler parameters. Once again, we fix the energy budget and hence the product $N_Y \times N_{\text{MALA}}$. In Table 16, we find that GMM40, and likely other multimodal targets, benefit monotonically from more particles and better mode coverage. Funnel instead suggests that targets with complex geometries need a minimum particle count for the auxiliary SMC approximation to be reliable and, once met, further MALA equilibration is more valuable than additional particles. We remark that these choices also depend on how finely discretised the diffusion process is. In the event where the discretised process takes large steps, more MALA steps are required for particles to adapt to the likewise rapidly changing posterior. In Table 17, we find reasonable choices for the ESS threshold κ have no major effect on sample quality for these targets.

Table 15: Sensitivity of DPSMC (OU) to LL initialisation parameters λ_0 and N_{LL} across three seeds. Omitted ESS error bars are at most ± 0.1. Best result for each metric under each target is in **bold**.

GMM40 ($d = 2$)										
	MTVD (↓)					ESS (↑)				
λ_0 / N_{LL}	0	16	32	64	128	0	16	32	64	128
0.167	$0.040_{\pm 0.005}$	$0.034_{\pm 0.004}$	$0.034_{\pm 0.002}$	$0.036_{\pm 0.002}$	$0.035_{\pm 0.004}$	**112.1**	105.4	104.8	103.8	102.3
0.333	$0.038_{\pm 0.005}$	$0.030_{\pm 0.005}$	$0.034_{\pm 0.002}$	$0.039_{\pm 0.001}$	$0.038_{\pm 0.004}$	109.5	102.8	102.2	101.7	101.3
0.5	$0.036_{\pm 0.001}$	$\mathbf{0.029_{\pm 0.003}}$	$0.033_{\pm 0.003}$	$0.034_{\pm 0.002}$	$0.036_{\pm 0.002}$	107.7	102.2	101.6	101.3	100.5
0.667	$0.057_{\pm 0.003}$	$0.059_{\pm 0.008}$	$0.062_{\pm 0.006}$	$0.068_{\pm 0.010}$	$0.073_{\pm 0.010}$	106.0	101.4	100.8	100.2	99.1
0.833	$0.123_{\pm 0.005}$	$0.144_{\pm 0.007}$	$0.152_{\pm 0.009}$	$0.151_{\pm 0.002}$	$0.159_{\pm 0.004}$	104.8	99.8	99.2	98.4	97.3
Funnel ($d = 10$)										
	KS (↓)					ESS (↑)				
λ_0 / N_{LL}	0	16	32	64	128	0	16	32	64	128
0.167	$0.024_{\pm 0.001}$	$0.036_{\pm 0.003}$	$0.033_{\pm 0.003}$	$0.030_{\pm 0.000}$	$0.025_{\pm 0.000}$	**94.2**	92.7	92.1	91.0	89.0
0.333	$0.033_{\pm 0.001}$	$0.041_{\pm 0.002}$	$0.032_{\pm 0.002}$	$0.025_{\pm 0.001}$	$\mathbf{0.021_{\pm 0.001}}$	93.1	92.1	91.4	90.1	87.8
0.5	$0.059_{\pm 0.002}$	$0.047_{\pm 0.002}$	$0.029_{\pm 0.001}$	$0.028_{\pm 0.001}$	$0.034_{\pm 0.001}$	92.5	91.5	90.9	89.5	86.8
0.667	$0.090_{\pm 0.001}$	$0.043_{\pm 0.000}$	$0.051_{\pm 0.001}$	$0.065_{\pm 0.002}$	$0.072_{\pm 0.002}$	92.2	91.2	90.5	89.0	86.0
0.833	$0.135_{\pm 0.002}$	$0.082_{\pm 0.001}$	$0.106_{\pm 0.001}$	$0.130_{\pm 0.003}$	$0.141_{\pm 0.004}$	91.7	90.6	89.9	88.3	84.9

Table 16: Sensitivity of DPSMC (OU) to trading off particle count N_Y against MALA step count N_{MALA} under a fixed energy budget, across ten seeds. NESS is the normalised ESS, i.e. ESS$/N_Y$. Best result for each metric under each target is in **bold**, and second-best is underlined.

Target / (N_Y, N_{MALA})		$(4, 32)$	$(8, 16)$	$(16, 8)$	$(32, 4)$	$(64, 2)$	$(128, 1)$
GMM40 ($d = 2$)	MTVD (↓)	$0.069_{\pm 0.005}$	$0.049_{\pm 0.004}$	$0.042_{\pm 0.004}$	$0.040_{\pm 0.004}$	$\mathbf{0.037_{\pm 0.005}}$	$\mathbf{0.037_{\pm 0.004}}$
	NESS (↑)	$\mathbf{0.80_{\pm 0.00}}$	$0.77_{\pm 0.00}$	$0.77_{\pm 0.00}$	$0.77_{\pm 0.00}$	$0.78_{\pm 0.00}$	$\underline{0.79_{\pm 0.00}}$
Funnel ($d = 10$)	KS (↓)	$0.040_{\pm 0.004}$	$\mathbf{0.020_{\pm 0.001}}$	$\underline{0.029_{\pm 0.004}}$	$0.034_{\pm 0.003}$	$0.032_{\pm 0.002}$	$0.030_{\pm 0.002}$
	NESS (↑)	$\mathbf{0.76_{\pm 0.00}}$	$\underline{0.74_{\pm 0.00}}$	$0.72_{\pm 0.00}$	$0.71_{\pm 0.00}$	$0.70_{\pm 0.00}$	$0.70_{\pm 0.00}$

Table 17: Sensitivity of DPSMC (OU) to the ESS threshold κ, across ten seeds. Resampling is triggered only when the ESS falls below κN_Y.

Target / κ		0.3	0.5	0.7
GMM40 ($d = 2$)	MTVD (↓)	$0.035_{\pm 0.006}$	$0.033_{\pm 0.001}$	$0.036_{\pm 0.004}$
	ESS (↑)	$84.1_{\pm 0.2}$	$101.3_{\pm 0.1}$	$113.0_{\pm 0.0}$
Funnel ($d = 10$)	KS (↓)	$0.029_{\pm 0.002}$	$0.030_{\pm 0.002}$	$0.030_{\pm 0.002}$
	ESS (↑)	$73.3_{\pm 0.1}$	$89.5_{\pm 0.0}$	$102.6_{\pm 0.0}$

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We develop diffusion-based samplers by designing an SMC framework for score estimation and control variates for variance reduction. We also provide convergence guarantees for our algorithm and demonstrate its competitiveness on a range of benchmarks. Hence, our claims accurately reflect our paper's contribution and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss limitations of our work in our experiments and discussion sections.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We explicitly state assumptions for our theoretical analysis, i.e. Assumptions 1 and 2, and provide complete proofs in the Appendix for all stated results: Proposition 1 (Appendix B.1), Proposition 2 (Appendix C.2.1), Proposition 3 (Appendix C.2.3), Proposition 4 (Appendix F.1), and Theorem 1 (Appendix F.2).

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: We report all the hyperparameters for our method, including an explicit algorithm for how it works. Further, we provide code with sufficient instructions to faithfully reproduce the main experimental results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide code with sufficient instructions to faithfully reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We explicitly define the target distributions and the setup and hyperparameters of samplers considered in Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the standard error of benchmark metrics computed for our main results found in Tables 1 and 2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: We provide details pertaining to runtimes and specific GPU hardware used to run experiments in Table 12 of Appendix H.2.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in our paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our paper advances the field of machine learning by enhancing the understanding of sampling from unnormalised probability distributions. There are many potential societal impacts of our work, none of which we feel must be specifically highlighted here.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: Our paper does not use existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: Our paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.