AMORTISED INFERENCE THROUGH ONE-STEP IMPLICIT SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Efficient sampling from Bayesian posterior distributions in high dimensions remains a hard problem, despite advances in theory and algorithms. Multiple steps are usually required to draw unbiased samples, for example, iterations of an MCMC algorithm or integration steps in a diffusion-based amortised sampler. However, scalable one-step amortised inference remains a challenging open problem. In this work, we build upon recent progress in data-driven generative modelling to offer a possible solution for one-step amortised sampling. Our *one-step implicit sampler* (OSIS) uses short-time MCMC dynamics to define a local target for the evolution of an implicitly modelled distribution. In the asymptotic limit, the modelled distribution is a fixed point of the MCMC kernel and thus samples from the target. We analyse the convergence properties of OSIS and show connections with other generative modelling and sampling algorithms. We study various design choices and show that OSIS performs well compared to multi-step amortised samplers and non-amortised MCMC baselines while being far more efficient at sampling time.

1 INTRODUCTION

The sampling problem. Many problems in physics, biology, statistics, and machine learning (Welling & Teh, 2011; Wirnsberger et al., 2020; Noé et al., 2019; Holdijk et al., 2023) require drawing samples from a distribution whose density is known up to a normalising constant. Formally, this means sampling from a distribution defined via an *energy function* $\mathcal{E} : \mathbb{R}^d \to \mathbb{R}$, with density

$$\pi(x) = e^{-\mathcal{E}(x)}/Z, \quad Z = \int_{\mathbb{R}^d} e^{-\mathcal{E}(x)} \, \mathrm{d}x. \tag{1}$$

A typical instance of this problem is Bayesian inference: a posterior $\pi(x \mid y) \propto p(x)p(y \mid x)$ over a variable x given an observation y has the form in (1) with $\mathcal{E}(x) = -\log p(x) - \log p(y \mid x)$.

The classic way to sample from a distribution as in (1) is by using (training-free) Monte Carlo algorithms. Many such algorithms draw samples by simulating an ergodic chain using π-invariant kernels for a sufficiently long time. Popular Monte Carlo algorithms include Metropolis-Hastings (Metropolis et al., 1953; Hastings, 1970), unadjusted and Metropolis-adjusted Langevin (Rossky et al., 1978; Besag, 1994; Roberts & Tweedie, 1996), Hamiltonian Monte Carlo (Duane et al., 1987; Neal, 2011), and sequential Monte Carlo (Del Moral et al., 2006). The use of these methods comes at the cost of running chains for many steps, or with many simulated particles, to obtain high-quality samples from the target.

Generative modelling algorithms that learn to imitate unbiased samples from a distribution – *i.e.*, approximate an empirical measure – scale quite successfully, as evidenced by many large-scale applications of deep learning (Kong et al., 2021; Hoogeboom et al., 2022; Sahoo et al., 2024). Meanwhile, generative modelling algorithms that rely only on a density function, without unbiased samples, continue to lag behind, and scaling them remains a challenge despite the deep significance of this problem.

Amortised sampling. Deep learning has brought about new possibilities for introducing learnable components into sampling algorithms. Such components allow to spend more time and compute for training, yet gain speed when sampling and exploit generalisable structure in the target. Amortisation techniques include learning MCMC transition kernels (Song et al., 2017; Samsonov et al., 2022; Wang et al., 2025), amortising MCMC dynamics (Li et al., 2017), and training density estimators to sample from unnormalised distributions (Rezende & Mohamed, 2015; Midgley et al., 2023).

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

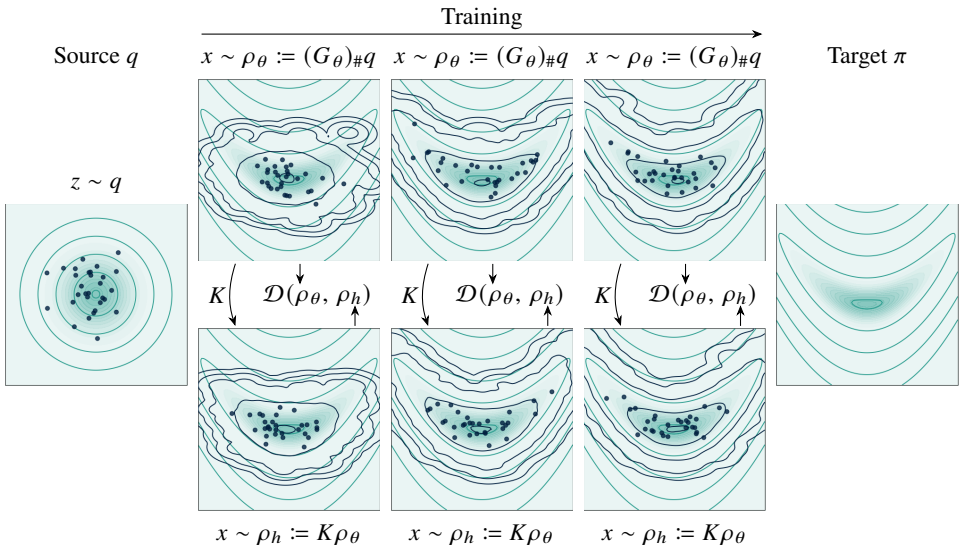

Figure 1: **OSIS evolves an implicit modelled distribution ρ_θ towards a target π available through its unnormalised density (green contours).** G_θ maps samples z the source distribution q to samples $x = G_\theta(z)$ from the modelled distribution $\rho_\theta := (G_\theta)_\# q$ (blue KDE contours). This pushforward distribution is evolved by a π-invariant MCMC kernel K, giving $\rho_h := K\rho_\theta$. The parameters θ are updated to minimise an empirical divergence between ρ_θ and ρ_h. Upon convergence, $\rho_\theta \approx \pi$.

The most successful family of amortised samplers adopts diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) as approximators of the target distribution (*e.g.*, Zhang & Chen, 2022; Vargas et al., 2023; Richter et al., 2024; Sendera et al., 2024; Chen et al., 2025a; Albergo & Vanden-Eijnden, 2025; Berner et al., 2026; Havens et al., 2025). Diffusion samplers introduce a learnable stochastic process to model the transition dynamics from a known distribution to a target, usually by approximately reversing a fixed noising process. In contrast to diffusion models trained using a variational bound on the likelihood of empirical samples, diffusion samplers are trained using stochastic optimal control objectives (Zhang & Chen, 2022; Berner et al., 2024) or path-space divergences (Richter et al., 2020; Lahlou et al., 2023), among other choices. Recent work has found and exploited connections between diffusion samplers and amortisation of SMC (Choi et al., 2026).

One-step amortised sampling. Although diffusion samplers allow for a considerable inference speed-up by avoiding long-chain MCMC, they require simulation of a neural stochastic dynamical system during both training and sampling. Amortised sampling with a *single* neural function evaluation, however, remains a challenge. Although it is possible to use normalising flows for one-step sampling (Rezende & Mohamed, 2015), more recent work shows that such methods scale poorly and are prone to mode collapse (He et al., 2025). These methods can further be placed within a Monte Carlo outer loop and augmented with MCMC sampling (Midgley et al., 2023).

In an attempt to design an efficient one-step sampler, we take a different approach, drawing inspiration from the novel paradigm in generative modelling termed drifting models (Deng et al., 2026; Gretton et al., 2026). Drifting models evolve a implicit distribution towards an empirical target by defining a local direction for its evolution via a *drifting field* estimated from batches of target and generated samples. Such an approach, as previously formulated, cannot be directly applied to sampling, as it requires access to samples from the target distribution to compute the drifting field.

Our contributions. We bypass this limitation by using an MCMC kernel invariant to the target distribution to implicitly define a drifting field. This leads us to the first main contribution:

(1) We propose an algorithm for training a one-step implicit sampler (OSIS) using an objective that enforces invariance of the sampler's output distribution to a target-invariant MCMC kernel (§2.1).

There have been prior attempts to design a one-step amortised sampling algorithm, using kernel density approximations to circumvent data unavailability (Cao et al., 2026) and using adversarial training (Li et al., 2017). We discuss the connections between OSIS, drifting models, computational

Wasserstein gradient flows, and this prior work in §3. We also study the theoretical properties of OSIS, showing global convergence under assumptions similar to those in the MCMC literature:

(2) We prove convergence of OSIS under asymptotic conditions and constraints on the target (§4).

To achieve scalability and efficiency in practice, we study the empirical performance of OSIS under different design choices, such as choice of MCMC kernel, empirical divergence, and other techniques:

(3) We study design choices for OSIS (§5) and their influence on performance (Table 3).

Finally, we benchmark our sampler on amortised inference problems ranging from synthetic targets to posterior sampling under large-scale image model priors, leading to our final contribution:

(4) We show that OSIS performs on par with multi-step baselines, while being more efficient (§6.1) and remaining scalable (§6.2).

2 Proposed method: Training by distilling MCMC refinements

2.1 Preliminaries

Problem setting: one-step amortised sampling. Suppose that $G : \mathbb{R}^{d_z} \to \mathbb{R}^d$ is a measurable function and q is a simple distribution (*e.g.*, $\mathcal{N}(0, I)$) over $\mathbb{R}^{d_z}$, where d_z is not necessarily equal to d. Define a random variable $z \sim q$ and $x = G(z)$. The distribution of x is the pushforward of q by G, denoted $\rho = G_{\#}q$. If G is not (efficiently) invertible or differentiable, the density of ρ is intractable to compute. Given a target distribution π over $\mathbb{R}^d$ defined by an unnormalised density as in (1), the goal of one-step amortised sampling is to find G such that $\pi \approx G_{\#}q$. If the function G depends on parameters θ, we denote it G_θ and abbreviate $\rho_\theta = (G_\theta)_{\#}q$. The goal is then to find θ such that $\rho_\theta \approx \pi$.

Algorithm 1 OSIS using a divergence $\mathcal{D}$

Require: Prior distribution q, π-invariant kernel K, generator G_θ, divergence $\mathcal{D}$.

```
while not converged do
    ρ_θ ← G_θ#q
    ▷ Apply the MCMC kernel
    ρ_h ← K ρ_sg[θ]
    ▷ Compute loss using D
    L(θ) ← D(ρ_θ, ρ_h)
    Gradient update on θ with ∇_θ L
end while
return G_θ
```

Stationary kernels. Suppose that K is a stochastic kernel from $\mathbb{R}^d$ to $\mathbb{R}^d$ (*i.e.*, a conditional distribution $K(y \mid x)$ satisfying measurability conditions). One can construct a Markov chain $(x_k)_{k=0}^{\infty}$ via $x_{k+1} \sim K(x_{k+1} \mid x_k)$. Markov chain Monte Carlo algorithms select K such that, regardless of the distribution of x_0, the chain converges to a unique stationary distribution π with respect to which K is invariant, *i.e.*, $K\pi(\mathrm{d}y) := \int K(\mathrm{d}y \mid x)\pi(x)\,\mathrm{d}x = \pi(\mathrm{d}y)$ (Douc et al., 2018). Many choices for K use a discrete-time approximation to Langevin dynamics, such as ULA, MALA (Rossky et al., 1978; Besag, 1994; Roberts & Tweedie, 1996), HMC (Duane et al., 1987; Neal, 2011), and improved Langevin integrators (Leimkuhler & Matthews, 2015).

2.2 Training objectives for one-step sampling

Training of a one-step sampler, by some algorithm, produces a sequence of parameter iterates $(\theta_k)_{k\geq 0}$, and thus an evolution of the modelled distribution $\rho_{\theta_k} := (G_{\theta_k})_{\#}q$. The training algorithm should be designed to ensure that ρ_{θ_k} converges to π as $k \to \infty$. One way to express such convergence is through invariance of the limiting distribution with respect to a π-invariant stochastic kernel K.

Divergence-based loss. Given a π-invariant kernel K, a one-step sampler $\rho_\theta = (G_\theta)_{\#}q$ should satisfy $\rho_\theta \approx K\rho_\theta$ upon convergence. We propose to turn this into a training objective:

$$\mathcal{L}(\theta) = \mathcal{D}\left(\rho_\theta,\, K\rho_{\mathrm{sg}[\theta]}\right), \tag{2}$$

where $\mathcal{D}$ is some measure of divergence between distributions, $K\rho(\mathrm{d}y) := \int K(\mathrm{d}y \mid x)\rho(\mathrm{d}x)$, and $\mathrm{sg}\,[\cdot]$ is the stop-gradient operator.

The complete algorithm, given in Algorithm 1, leaves room for two design choices: the kernel K and the divergence $\mathcal{D}$. Both should be efficient to compute, and the distance $\mathcal{D}$ should be differentiable with respect to (empirical approximations of) ρ_θ. We discuss these choices in §C.1 and §C.2, respectively. Additional aspects of training are discussed in §5 and variants of Algorithm 1 in §A.

3 Connections and related work

Connection to other implicit samplers. The connection between generative modelling and implicit models, those without a parametrisation of density, goes back to Mohamed & Lakshminarayanan (2016). Subsequent works explored algorithms that use MCMC to train implicit samplers: Feng et al. (2017) amortises Stein variational gradient descent (SVGD; Liu & Wang, 2016), while Titsias (2017) fits an invertible map to sample from the distribution induced by a short MCMC chain. Related approaches include Hoffman (2017), Ruiz & Titsias (2019), and Gabrié et al. (2022).

The closest precursor to OSIS is Li et al. (2017). That algorithm initialises an MCMC chain from an implicit approximation network, applies a transition kernel, and updates the network with an adversarial loss matching the initial samples to their few-step refinements. OSIS differs from this line of work in three key ways. First, unlike Li et al. (2017), OSIS does not rely on an adversarial loss, but remains stable with a fixed choice of empirical divergence. Second, OSIS is compatible with any MCMC kernel and any divergence between distributions, and we compare several choices of each (see Table 3 for kernels and Table 1 for divergences). Third, we link the convergence of these methods to the convergence of the underlying MCMC kernel and establish a connection to Wasserstein gradient flows that was not known before in the sampling context (see §4).

Connection to drifting models. OSIS can be linked to drifting models (Deng et al., 2026), which define a local target for the evolution of a modelled distribution to guide it towards a target given empirically through data samples. Although OSIS and drifting models solve different problems (approximating an empirical data distribution and an energy-based distribution, respectively), they share some similarities. In Deng et al. (2026), stationarity of an implicitly defined generative model is expressed through a drifting field, a vector field defining an evolution of particles under which the target distribution is stationary under asymptotic conditions. The field is defined similarly to the gradient of a kernel-based divergence and thus evolves model-generated samples towards the data distribution over the course of training. Unlike in diffusion and flow-based models, which approximate a time-dependent evolution, this field is a non-stationary training target, rather than part of the generation process.

More recent works propose other ways to define this drifting field, some of which can be identified with gradients of divergences between distributions, making the training process simulate a gradient flow (He et al., 2026; Gretton et al., 2026; Han et al., 2026). As we discuss below, OSIS shares parts of this connection; however, drifting models and their successors *approximate the vector field giving the gradient flow*, whereas OSIS *uses the vector field giving the gradient flow to define the local target for optimisation*.

Cao et al. (2026) proposes a modification of drifting models that makes them applicable to sampling; the modification relies on kernel density estimation to model the score of the target distribution. They directly employ local energy-based importance sampling or a second-order curvature approximation to derive a target-side drift. In contrast, OSIS defines an implicit drift using the gradient of a discrepancy functional toward an MCMC-refined distribution.

Connection to computational Wasserstein gradient flows. A Wasserstein gradient flow (WGF) describes a continuous evolution $(\rho_t)_{t\geq 0}$ on the space of absolutely continuous probability distributions with finite second moments $\mathcal{P}_2(\mathbb{R}^d)$. The flow moves a distribution ρ along the gradients of a functional $\mathcal{J} : \mathcal{P}_2(\mathbb{R}^d) \to \mathbb{R}$. The flow admits two equivalent descriptions: as an evolution of particles $x_t \sim \rho_t$ (3) and the corresponding continuity equation, the evolution of the density ρ_t (4):

$$\frac{\mathrm{d}}{\mathrm{d}t}x_t = -\nabla\frac{\delta\mathcal{J}}{\delta\rho_t}(x_t), \tag{3}$$

$$\frac{\partial}{\partial t}\rho_t = \nabla\cdot\left(\rho_t\,\nabla\frac{\delta\mathcal{J}}{\delta\rho_t}\right), \tag{4}$$

where initial conditions are given by $x_0 \sim \rho_0$ and $\frac{\delta\mathcal{J}}{\delta\rho} : \mathbb{R}^d \to \mathbb{R}$ denotes the first variation of $\mathcal{J}$. The flow given in (4) admits a discrete-time approximation scheme by proximal steps in the Wasserstein geometry, known as the Jordan-Kinderlehrer-Otto (JKO) scheme (Jordan et al., 1998):

$$\rho_{k+1} = \operatorname*{argmin}_{\rho}\left\{2\eta\mathcal{J}(\rho) + \mathcal{W}_2^2(\rho,\rho_k)\right\} \tag{5}$$

that converges to the continuous-time flow as $\eta \to 0$. Refer to Ambrosio et al. (2005) for a rigorous treatment or to Santambrogio (2017) for a succinct introduction.

The scheme in (5) offers a tempting way to solve optimisation problems on the space of probability distributions, and recent years have seen multiple attempts to convert it into a scalable machine learning algorithm. Mokrov et al. (2021) proposes to learn WGF by parametrising each step of the JKO scheme (5) via an input-convex neural network. An alternative approach was introduced in Galashov et al. (2025). Some more recent works (Chen et al., 2025b; Tian et al., 2026) offer ways to compute WGF using different variations of MMD as the functional $\mathcal{J}$. Finally, Gretton et al. (2026) provides an interpretation of drifting (§3) as WGF. He et al. (2026); Han et al. (2026) show that Sinkhorn divergence to the target distribution can be used as the functional and can achieve strong results on image generation. Notably, *all methods that attempt to neurally approximate WGF to this day fundamentally rely on the availability of unbiased samples*.

4 Theoretical analysis of OSIS

In this section we analyse the convergence of OSIS under idealised assumptions:

(1) We take K to be continuous Langevin diffusion and $\mathcal{D}$ to be $\frac{1}{2}\mathcal{W}_2^2$. In practice, this assumption is satisfied in the infinite-particle limit of empirical estimation of the divergence and with perfect numerical integration in the Monte Carlo updates.
(2) We assume that optimisation of the modelled distribution is performed in Wasserstein space, rather than the parameter space (θ). This can be achieved in practice by replacing the gradient step on $\mathcal{W}_2^2$ with respect to θ by a proximal gradient step in the $\mathcal{W}_2^2$ geometry and assuming the parametric family and optimisation algorithm are sufficiently expressive to perform this step exactly.

Our analysis uses that the evolution of density given by Langevin dynamics is the WGF of the KL divergence functional; we show that OSIS inherits the convergence properties of Langevin dynamics (*e.g.*, Durmus & Moulines, 2017; Chewi et al., 2025).

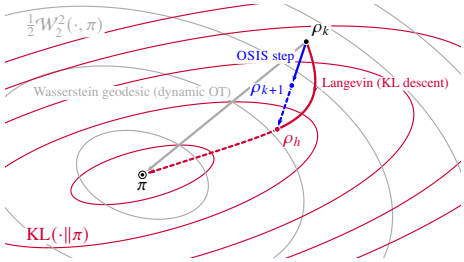

Figure 2: One idealised OSIS iteration. KL and Wasserstein distance share the minimiser π. However, KL decreases along the Langevin path (red), whereas Wasserstein decreases along the geodesic (grey). OSIS takes a step along the Wasserstein gradient of $\frac{1}{2}\mathcal{W}_2^2(\cdot,\rho_h)$ towards ρ_{k+1} (blue).

4.1 Setting

We first formalise the algorithm described in §2.1.

Notation and assumptions. We assume that all distributions are elements of $\mathcal{P}_2(\mathbb{R}^d)$, the space of absolutely continuous distributions on $\mathbb{R}^d$ with finite second moment, and we identify them with their densities. We assume that the target $\pi \propto e^{-\mathcal{E}}$ is defined by a twice continuously differentiable energy $\mathcal{E} : \mathbb{R}^d \to \mathbb{R}$. Let $\{X_t\}_{t\geq 0}$ be a Langevin process, defined by an Itô SDE:

$$\mathrm{d}X_t = -\nabla\mathcal{E}(X_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t, \tag{6}$$

where $\{B_t\}_{t\geq 0}$ is d-dimensional Brownian motion. For $h \geq 0$, we define K^h to be the stochastic integration kernel of the process (6), *i.e.*, $K^h(A \mid x) := \mathbb{P}(X_{t+h} \in A \mid X_t = x)$, which is π-invariant.

For a distribution ρ and an absolutely continuous kernel K we denote by $K\rho$ the pushforward distribution, which has density $K\rho(y) := \int K(y \mid x)\rho(x)\mathrm{d}x$. As before, we denote by $T_\#\rho$ the pushforward of a distribution ρ by a measurable function T. Integration is assumed to be taken over the entire $\mathbb{R}^d$ unless stated otherwise. We use $|\cdot|$ to denote Euclidean distance and define 2-Wasserstein distance and KL divergence between distributions ρ and μ as follows:

$$\mathcal{W}_2^2(\rho,\mu) = \inf_{T_\#\rho=\mu}\left\{\int |T(x)-x|^2\rho(x)\mathrm{d}x\right\}, \quad \mathrm{KL}\,(\rho\,\|\,\mu) = \int \log\frac{\rho(x)}{\mu(x)}\rho(x)\mathrm{d}x.$$

We use the Monge formulation of 2-Wasserstein distance, since for absolutely continuous distributions, the Monge optimal transport maps exist and are unique. Other formulations are discussed in §B.1.

One step of OSIS in Wasserstein space. We now formalise one step of OSIS, visualised in Figure 2. At each training iteration $k \geq 0$, OSIS has the initial distribution $\rho_k := \rho_{\theta_k} = (G_{\theta_k})_\# q$.

First, we use $\rho_{h_k} := K^{h_k}\rho_k$ to denote the distribution after a continuous evolution of ρ_k by a stochastic kernel K^{h_k} for the time $h_k > 0$. **Second**, we compute the 2-Wasserstein OT map T^{h_k}

between ρ_k and ρ_{h_k}. With ρ_{h_k} fixed, the functional $\frac{1}{2}\mathcal{W}_2^2(\cdot, \rho_{h_k})$ has Wasserstein gradient $\mathrm{Id} - T^{h_k}$ at ρ_k (Alfonsi & Jourdain, 2020, Theorem 2.2). **Third**, we set $T_{\eta_k}^{h_k} := \eta_k T^{h_k} + (1-\eta_k)\,\mathrm{Id}$ with $\eta_k \in (0,1]$, which is the result of taking a gradient step on $\frac{1}{2}\mathcal{W}_2^2(\cdot, \rho_{h_k})$ with step size η_k. Finally, one idealised iteration of OSIS is expressed as

$$\rho_{k+1} := (T_{\eta_k}^{h_k})_{\#}\rho_k. \tag{7}$$

4.2 Main results

We prove convergence of the procedure (7) under various assumptions on π. Proofs appear in §B.2.

Convergence for strongly log-concave target. We consider the case where π is λ-strongly log-concave (λ-SLC), that is, $\nabla^2\mathcal{E} \succeq \lambda I$ for some $\lambda > 0$.

Theorem 4.1 (Contraction of one step of OSIS under λ-SLC π)**.** *Let π be λ-SLC. Let $\rho_k, \rho_{h_k}, T^{h_k}, T_{\eta_k}^{h_k}, \rho_{k+1}$ be as in §4.1. For any $h_k > 0$, $\eta_k \in (0,1]$, the following bounds hold:*

$$\mathcal{W}_2(\rho_{k+1}, \pi) \le \left((1-\eta_k)(1+e^{-\lambda h_k}) + e^{-\lambda h_k}\right)\mathcal{W}_2(\rho_k, \pi); \tag{8}$$

$$\mathrm{KL}\left(\rho_{k+1} \,\|\, \pi\right) \le \left(1 - \eta_k(1 - e^{-2\lambda h_k})\right)\mathrm{KL}\left(\rho_k \,\|\, \pi\right). \tag{9}$$

Theorem 4.2 (Convergence of OSIS under λ-SLC π)**.** *Under the assumptions of Theorem 4.1, let $(\rho_k)_{k\ge 0}$ be the sequence of probability distributions produced by the* OSIS *iterations. Assuming $h_k > 0$ and $\eta_k \in (0,1]$ for all $k \ge 0$, the following bounds hold for any $N \ge 1$:*

$$\mathrm{KL}\left(\rho_N \,\|\, \pi\right) \le e^{-\sum_{k=0}^{N-1}\eta_k(1-e^{-2\lambda h_k})}\,\mathrm{KL}\left(\rho_0 \,\|\, \pi\right). \tag{10}$$

Moreover, if $\bar{h} = \inf_{k\ge 0} h_k > 0$, $\bar{\eta} = \inf_{k\ge 0}\eta_k > 0$, and $\bar{\eta} > 2/(1+\exp(\lambda\bar{h}))$, then:

$$\mathcal{W}_2(\rho_N, \pi) \le \left((1-\bar{\eta})(1+e^{-\lambda\bar{h}}) + e^{-\lambda\bar{h}}\right)^N \mathcal{W}_2(\rho_0, \pi); \tag{11}$$

As a result, we have shown a geometric convergence rate for OSIS under a λ-SLC assumption on π.

Beyond strong log-concavity. The same update also converges for non-log-concave π satisfying an α-log-Sobolev inequality (α-LSI) and a lower curvature bound $\nabla^2\mathcal{E} \succeq -\kappa I$. Here $\alpha > 0$ and α-LSI means $2\alpha\,\mathrm{KL}\left(\rho \,\|\, \pi\right) \le \int |\nabla \log(\rho/\pi)|^2 \rho\,\mathrm{d}x$ for all ρ. In the appendix (Theorem B.5), we prove convergence rates, which in particular imply:

Corollary 4.3 (Geometric convergence under LSI and bounded negative curvature)**.** *Under the notations of §4.1, and the α-LSI and curvature assumptions above, with fixed $h_k = h > 0$ and $\eta_k = \eta \in (0,1]$ with $\kappa h(1-\eta) < 2$, $\mathcal{W}_2^2(\rho_k, \pi)$ and $\mathrm{KL}\left(\rho_k \,\|\, \pi\right)$ converge geometrically to 0.*

5 Practical considerations

The idealised setting, which is detailed in Algorithm 1 and theoretically analysed in §4 cannot be implemented in practice. Instead, we approximate the distributions and the divergence between them 8by their empirical counterparts and replace the continuous-time Langevin by a discrete-time MCMC simulation. The next paragraphs detail various design choices for OSIS relevant in practice. An extended introduction to the practical considerations of the algorithm is provided in §C.

Refinement kernel. OSIS can be instantiated with different MCMC kernels. The theory concerns exact Langevin, but we use the MALA kernel in practice and consider other choices in Table 3.

Empirical divergence. It is not possible to use infinite populations as considered in the theory, instead OSIS computes finite-batch empirical divergences between generated samples and their kernel-refined versions. Our presented theory considers a squared 2-Wasserstein loss, but OSIS can be extended to other discrepancies between distributions. For the experiments on the synthetic targets (§6.1) we use sliced Wasserstein distances (SWD; Bonneel et al., 2015), and for the experiments on Bayesian posteriors (§6.2) we use minibatch OT with squared-Euclidean cost, which converges to exact 2-Wasserstein in the limit (Fatras et al., 2021).

Annealing. Kernel refinements can initially target the tempered distribution $\pi_\beta \propto \exp(-\beta\mathcal{E})$ with β_t and then anneal β to 1 over the course of training. The starting temperature and annealing schedule balance early exploration with learning the final target.

Table 1: Sampler comparison on GMM, Funnel, and ManyWell targets. Entries are mean ± standard deviation over five seeds. Second column shows number of NN and energy evaluations per sample

Target density →		40GMM ($d = 2$)		40GMM ($d = 10$)		Funnel ($d = 10$)		ManyWell ($d = 32$)		ManyWell ($d = 64$)	
Method ↓ Metric → NN / $\mathcal{E}$		$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)
Reference		7.68±3.22	7.17±1.51	202.81±16.77	7.98±0.78	87.60±1.20	9.22±1.54	23.61±0.07	8.75±0.72	78.36±0.11	9.58±0.65
OSIS	1 / 0	103.26±41.41	64.37±13.38	3086.96±439.59	139.26±36.09	93.99±3.33	28.06±3.06	26.79±0.41	55.92±6.00	75.88±0.34	91.05±3.98
FAB (flow only)	1 / 0	296.01±333.93	81.05±38.73	6217.94±472.86	382.64±44.65	583.48±233.71	12.94±2.34	27.10±0.46	47.95±4.56	125.58±0.04	401.28±0.64
FAB	1 / 120	72.03±19.29	46.42±6.32	6257.04±455.10	380.98±45.19	582.97±233.17	11.87±1.74	25.52±0.30	36.63±5.56	132.55±0.48	299.11±1.51
SMC	0 / 3825	14.67±1.79	16.85±2.11	4362.81±139.06	208.62±11.73	149.56±1.37	125.72±8.99	26.04±0.16	22.48±1.10	85.76±0.29	42.63±1.36
DIS	128 / 128	29.26±7.00	34.71±10.99	1650.04±60.09	85.37±5.26	122.58±21.01	214.06±103.49	36.44±2.44	162.78±20.94	100.34±3.52	162.04±18.66
PIS	128 / 128	40.35±25.29	34.40±12.85	2702.24±66.69	142.54±5.24	165.77±1.72	194.21±3.72	35.47±0.10	227.64±1.74	83.39±0.13	230.37±1.05
TB+MALA	128 / 128	10.20±4.11	9.50±2.66	408.80±70.18	17.38±2.13	508.96±66.54	106.16±3.89	23.71±0.18	15.31±2.42	78.48±0.37	21.59±1.97
SubTB+SMC	128 / 128	12.67±3.94	10.66±2.84	314.98±23.25	14.87±3.26	498.94±61.84	14.03±5.65	23.75±0.12	13.86±1.33	76.54±1.23	31.55±9.21

Complete algorithm. To facilitate training stability, we make several gradient steps after each MCMC refinement. The complete training procedure detailing the use of empirical divergence, finite-step MCMC kernel, annealing, and multiple gradient steps per iteration is presented in Algorithm 3.

6 Experiments

6.1 Synthetic benchmarks

Target distributions. We evaluate OSIS on standard sampling benchmarks: Neal's funnel ($d = 10$) (Neal, 2003), ManyWell ($d = 32, 64$) (Noé et al., 2019), and 40GMM ($d = 2, 10$) (Blessing et al., 2024). We also consider benchmarks in higher dimension: NICE-MNIST ($d = 196$) and NICE-FMNIST ($d = 728$) (Blessing et al., 2024).

Baselines. We evaluate OSIS against several amortised samplers: the normalising-flow-based FAB (Midgley et al., 2023) and diffusion samplers trained with different algorithms – PIS (Zhang & Chen, 2022), DIS (Berner et al., 2024), and off-policy methods TB+MALA (Sendera et al., 2024) and SubTB+SMC (Choi et al., 2026). Since FAB can either be used as a one-step sampler or placed in an annealed importance sampling loop, we compare OSIS to the one-step version of FAB as well. Additionally, we compare OSIS to SMC (Del Moral et al., 2006), a training-free sampling algorithm.

Table 2: Sampler comparison on NICE targets. Conventions follow Table 1.

Target density →		NICE-MNIST ($d = 196$)		NICE-FMNIST ($d = 784$)	
Method ↓ Metric → NN / $\mathcal{E}$		$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)
Reference		129.56±0.22	8.75±0.08	549.42±1.35	9.24±1.09
OSIS	1 / 0	189.59±20.34	184.55±68.71	1743.96±180.58	382.13±59.92
FAB (flow only)	1 / 0	162.20±1.91	57.94±11.09	1768.75±335.78	430.25±72.05
FAB	1 / 120	143.83±3.10	80.83±14.39	1746.47±335.35	438.79±71.94
SMC	0 / 3825	268.32±4.16	246.17±13.54	2856.36±15.26	629.18±3.24
DIS	128 / 128	307.73±55.89	208.21±22.97	5321.68±2656.55	421.53±124.32
PIS	128 / 128	237.63±3.08	225.89±16.69	2793.73±19.11	290.08±4.09
TB+MALA	128 / 128	266.44±7.03	329.57±14.95	3074.87±23.88	430.23±4.81
SubTB+SMC	128 / 128	285.59±80.69	296.43±144.70	4035.49±82.91	531.04±15.03

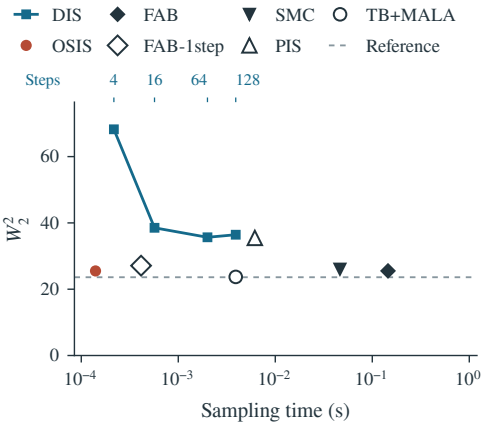

Figure 3: Sampling time comparison on ManyWell (median of 250 runs). DIS is trained for different numbers of denoising steps. OSIS achieves $\mathcal{W}_2^2$ comparable to the best baselines while sampling 10 to 1000 times faster.

Results. Table 1 shows sample-based metrics on synthetic targets, with 2-Wasserstein and MMD evaluated between 10000 draws from a trained OSIS model and an equal number of oracle samples from the target. OSIS is not consistently outperformed by any of the MCMC, normalising flow, or diffusion-based baselines despite requiring only one neural function evaluation and no energy evaluations to draw a sample. While the diffusion-based samplers trained by off-policy methods (Sendera et al., 2024; Choi et al., 2026) are the strongest overall, they require many sampling steps to perform well, *cf.* the analysis in Berner et al. (2026). On the higher-dimensional targets – given by normalising flow densities trained on MNIST and FMNIST treated as energy-based models – OSIS again performs on par with FAB and stronger than diffusion-based samplers (Table 2).

Figure 4 shows samples from various models on 40GMM ($d = 2$), Funnel ($d = 10$), ManyWell ($d = 32$), and MNIST ($d = 2$). It is clear from the figure that OSIS performs comparably in terms of sampling quality while being significantly faster at inference.

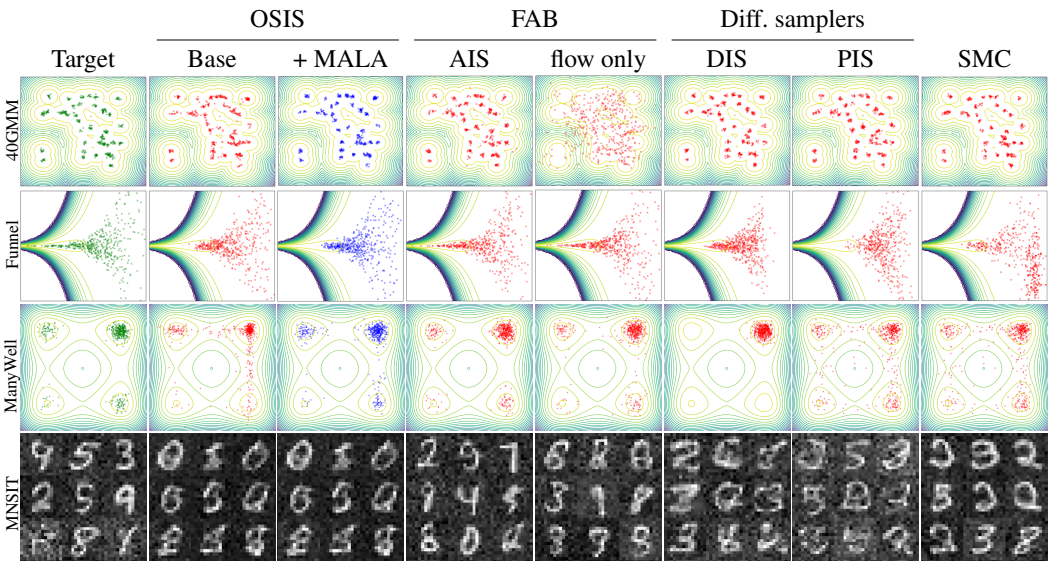

Figure 4: Visualisation of samples generated by OSIS and the baselines for 40GMM ($d = 2$), Funnel ($d = 10$), ManyWell ($d = 32$), MNIST ($d = 196$). Funnel and ManyWell are projected onto $\mathbb{R}^2$.

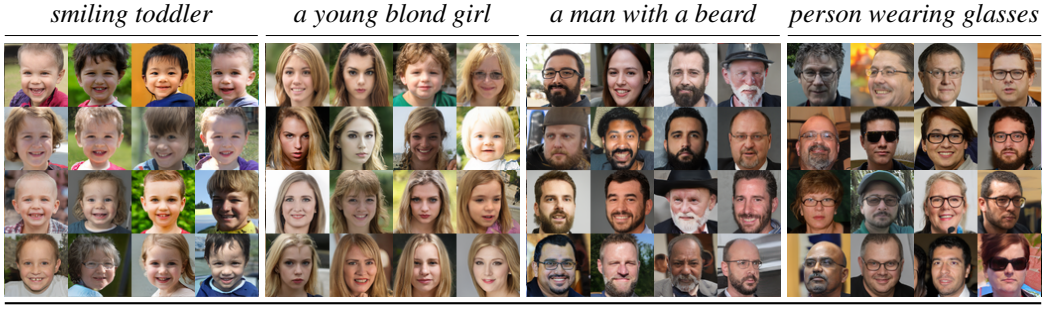

Figure 5: Uncurated samples obtained using outsourced sampling (§6.2). The samplers correspond to the ImageReward-based target conditioned on the given prompts. More examples in §E.6.

Sampling efficiency. The main advantage of one-step generative models is low sampling cost. OSIS, a one-step sampler, is much more efficient at sampling time than most baselines, whose sampling procedures require hundreds of function evaluations. To quantify this exactly, in Figure 3 we show sampling time per batch of 2000 samples on ManyWell ($d = 32$), measured on an NVIDIA GH200 GPU, excluding compilation and warm-up. OSIS yields a significant cost reduction with comparable or improved metrics (Table 1). To assess the trade-off between number of sampling steps and sampling quality, we also train and evaluate the diffusion-based sampler DIS with 4, 16, 64, and 128 denoising steps. Although DIS performance improves with the number of denoising steps, OSIS consistently outperforms at strictly lower sampling time.

Ablation of kernel choice. Table 3 compares different choices of the kernel in OSIS (see §C.1 for a description of the kernels). MALA, ULA, and OBABO tend to preform the strongest, and the higher-order splitting schemes for underdamped Langevin (OBABO and BAOAB) often improve over MALA, approaching the diffusion sampler baselines. However, based on these experiments, OSIS does not consistently benefit from more sophisticated refinement schemes.

Table 3: OSIS kernel ablation. Detailed experiment procedure is provided in §D. This table was computed using $M = 1$ inner iteration (see Algorithm 3).

Target density →	40GMM ($d = 2$)		ManyWell ($d = 32$)	
Kernel ↓ Metric →	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)
Reference	$7.68_{\pm 3.22}$	$7.17_{\pm 1.51}$	$23.61_{\pm 0.07}$	$8.75_{\pm 0.72}$
RWMH	$90.29_{\pm 1.08}$	$49.88_{\pm 0.56}$	$30.08_{\pm 0.62}$	$112.31_{\pm 6.76}$
ULA	$85.42_{\pm 25.65}$	$60.98_{\pm 13.75}$	$25.07_{\pm 0.29}$	$39.54_{\pm 6.08}$
MALA	$103.26_{\pm 41.41}$	$64.37_{\pm 13.38}$	$26.79_{\pm 0.41}$	$55.92_{\pm 6.00}$
BAOAB	$40.24_{\pm 16.81}$	$41.66_{\pm 15.91}$	$43.88_{\pm 3.70}$	$227.46_{\pm 52.12}$
OBABO	$81.85_{\pm 31.72}$	$58.06_{\pm 15.95}$	$28.77_{\pm 0.53}$	$85.55_{\pm 8.86}$
HMC	$117.90_{\pm 26.08}$	$96.11_{\pm 18.25}$	$30.08_{\pm 2.55}$	$106.84_{\pm 61.35}$
SVGD	$463.12_{\pm 263.92}$	$260.72_{\pm 125.03}$	$38.57_{\pm 3.62}$	$269.97_{\pm 50.16}$

Additionally, we compare with the RWMH kernel, which does not use energy gradients. Although it does not provide the best performance, it remains competitive and can serve as an alternative when target-energy gradients are unavailable. Note that

these methods are compared in terms of OSIS training steps, not at matched computational cost, and different kernels require different numbers of energy and gradient evaluations per step. Further ablations are provided in §E.4.

6.2 Bayesian posterior sampling under deep generative model priors

We show the scalability of OSIS by applying it to the problem of sampling from a Bayesian posterior under a pretrained generative model prior in latent space, known otherwise as *outsourced sampling* (Venkatraman et al., 2025). Suppose that a pretrained generative model is expressed as a deterministic transformation of a latent variable: $x = g(z)$, where $z \sim q(z)$ follows a known base distribution. The model's distribution over x, $p = g_{\#}q$, is treated as a Bayesian prior. If r is a likelihood or reward function, we wish to sample from the posterior, or reward-tilted distribution (which has density proportional to $p(x)r(x)$ if p has a density).

It can be shown that this posterior is the pushforward by g of the latent-space distribution with density proportional to $q(z)r(g(z))$, reducing sampling from the posterior in x-space to sampling from this distribution over the latent z. Past work adopting such a formulation (Venkatraman et al., 2025; Tang et al., 2025; Kalaivanan et al., 2025; Om et al., 2025; Wang et al., 2026; Tamogashev & Whitammer, 2026) used various amortised and Monte Carlo methods to sample from the latent-space posterior; here, we train OSIS samplers.

We conduct two sets of experiments:

(1) with g the decoder of a VAE (Kingma & Welling, 2014) trained on MNIST with 16-dimensional latent, with a pretrained classifier's likelihood $p_{\text{classifier}}(c \mid x)$ used to define the reward;
(2) with g the generator of a R3GAN (Huang et al., 2024) trained on FFHQ-64 with 64-dimensional latent, with exponentiated ImageReward (Xu et al., 2023) as a reward, $r(x) = \exp(\beta R_{\text{ImageReward}}(x, c))$ for a given caption c.

Details are given in §A.

MNIST with classifier reward. For the experiments with a VAE pretrained on MNIST, we use a mixture of different classes as the reward: $r(x) = \sum_{c=0}^{9} w_c p(c \mid x)$. Samples are shown in §E.5.

FFHQ-64 with ImageReward. We sample from the reward-reweighted distribution, which is defined by ImageReward conditioned on a prompt. For these experiments, we consider the four prompts shown in Figure 5, which shows the decoded samples from a trained OSIS model. For the prompt *a young blond girl*, we compare OSIS to a diffusion sampler (Sendera et al. (2024), as used for this problem in Venkatraman et al. (2025)), to an MCMC algorithm (MALA with adaptive step size), and to unbiased samples obtained by an expensive rejection sampling; see Table 4. OSIS performs similarly – higher mean reward, lower FID – to the diffusion-based amortised baseline (TB+MALA) while requiring only one function evaluation per sample. OSIS further improves with MCMC refinement. Visualisations in §E.6 show comparable quality of samples.

Table 4: Comparison on the outsourced sampling problem.

Algorithm ↓ Metric →	$\mathbb{E}[\log r(x)]$ (↑)	FID (↓)
Rejection sampling	2.164	2.345
MALA	2.127	3.070
OSIS	1.387	8.965
OSIS + refinement	1.854	7.510
TB+MALA	0.992	4.267

7 Conclusion

This paper has proposed an algorithm for one-step amortised sampling based on matching the distribution induced by a generator with its MCMC refinement. With appropriate techniques to make the refinement and divergence optimisation practical, OSIS performs surprisingly well compared to strong samplers that require many neural net and energy evaluations to draw each sample.

The most immediate challenge is to improve the sampling quality on highly multimodal targets, where diffusion samplers have been successful due to the geometric mixing induced by the noise, which is absent with tempering alone. The theoretical analysis should also be extended to include temperature annealing and inexact integration of the dynamics within various refinement kernels.

AI USE STATEMENT

The main text was written with no AI assistance. Generative AI tools were used (1) for assistance with generation of Figure 2; (2) for verifying the correctness of the theory, which had first been written and verified manually by the authors; (3) for orthographic corrections to some parts of the Appendix. We take responsibility for all content in this work.

REFERENCES

Michael S. Albergo and Eric Vanden-Eijnden. NETS: A non-equilibrium transport sampler. *International Conference on Machine Learning (ICML)*, 2025.

Aurélien Alfonsi and Benjamin Jourdain. Squared quadratic Wasserstein distance: optimal couplings and Lions differentiability. *ESAIM: Probability and Statistics*, 2020.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer, 2005.

Christophe Andrieu and Johannes Thoms. A tutorial on adaptive MCMC. *Statistics and computing*, 2008.

Dominique Bakry and Michel Émery. Diffusions hypercontractives. *Séminaire de probabilités*, 1985.

Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *Transactions on Machine Learning Research (TMLR)*, 2024.

Julius Berner, Lorenz Richter, Marcin Sendera, Jarrid Rector-Brooks, and Nikolay Malkin. From discrete-time policies to continuous-time diffusion samplers: Asymptotic equivalences and faster training. *Transactions on Machine Learning Research (TMLR)*, 2026.

Julian Besag. Comments on "Representations of knowledge in complex systems" by U. Grenander and M.I. Miller. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 56: 591–592, 1994.

Denis Blessing, Xiaogang Jia, Johannes Esslinger, Francisco Vargas, and Gerhard Neumann. Beyond ELBOs: A large-scale evaluation of variational methods for sampling. *International Conference on Machine Learning (ICML)*, 2024.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and Radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 2015.

Wenhan Cao, Keyu Yan, and Lin Zhao. One-step sampler for Boltzmann distributions via drifting. *arXiv preprint arXiv:2603.17579*, 2026.

Junhua Chen, Lorenz Richter, Julius Berner, Denis Blessing, Gerhard Neumann, and Anima Anandkumar. Sequential controlled Langevin diffusions. *International Conference on Learning Representations (ICLR)*, 2025a.

Zonghao Chen, Aratrika Mustafi, Pierre Glaser, Anna Korba, Arthur Gretton, and Bharath K Sriperumbudur. (De)-regularized maximum mean discrepancy gradient flow. *Journal of Machine Learning Research*, 2025b.

Sinho Chewi, Murat A Erdogdu, Mufan Li, Ruoqi Shen, and Matthew S Zhang. Analysis of Langevin Monte Carlo from Poincaré to log-Sobolev. *Foundations of Computational Mathematics*, 2025.

Sanghyeok Choi, Sarthak Mittal, Víctor Elvira, Jinkyoo Park, and Esmeralda S. Whitammer. Reinforced sequential Monte Carlo for amortised sampling. *International Conference on Machine Learning (ICML)*, 2026.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Neural Information Processing Systems (NeurIPS)*, 2013.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2006.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.

Randal Douc, Eric Moulines, Pierre Priouret, and Philippe Soulier. *Markov Chains*. Springer, 2018.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, 1987.

Alain Durmus and Eric Moulines. Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *The Annals of Applied Probability*, 2017.

Kilian Fatras, Younes Zine, Szymon Majewski, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Minibatch optimal transport distances; analysis and applications. *arXiv preprint arXiv:2101.01792*, 2021.

Yihao Feng, Dilin Wang, and Qiang Liu. Learning to draw samples with amortized Stein variational gradient descent. *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and MMD using Sinkhorn divergences. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. *Journal of Machine Learning Research*, 2021.

Luigi Fogliani, Bruno Loureiro, and Marylou Gabrié. Annealing in variational inference mitigates mode collapse: A theoretical study on Gaussian mixtures. *arXiv preprint arXiv:2602.12923*, 2026.

Marylou Gabrié, Grant M Rotskoff, and Eric Vanden-Eijnden. Adaptive Monte Carlo augmented with normalizing flows. *Proceedings of the National Academy of Sciences*, 119(10):e2109420119, 2022.

Alexandre Galashov, Valentin De Bortoli, and Arthur Gretton. Deep MMD gradient flow without adversarial training. *International Conference on Learning Representations (ICLR)*, 2025.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 2012.

Arthur Gretton, Li Kevin Wenliang, Alexandre Galashov, James Thornton, Valentin De Bortoli, and Arnaud Doucet. On the Wasserstein gradient flow interpretation of drifting models. *arXiv preprint arXiv:2605.05118*, 2026.

Jiaqi Han, Puheng Li, Qiushan Guo, Renyuan Xu, Stefano Ermon, and Emmanuel J. Candès. One-step generative modeling via Wasserstein gradient flows. *arXiv preprint arXiv:2605.11755*, 2026.

W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. 1970.

Aaron Havens, Benjamin Kurt Miller, Bing Yan, Carles Domingo-Enrich, Anuroop Sriram, Brandon Wood, Daniel Levine, Bin Hu, Brandon Amos, Brian Karrer, Xiang Fu, Guan-Horng Liu, and Ricky T. Q. Chen. Adjoint sampling: Highly scalable diffusion samplers via adjoint matching. *International Conference on Machine Learining (ICML)*, 2025.

Jiajun He, Yuanqi Du, Francisco Vargas, Dinghuai Zhang, Shreyas Padhy, RuiKang OuYang, Carla Gomes, and José Miguel Hernández-Lobato. No trick, no treat: Pursuits and challenges towards simulation-free training of neural samplers. *Frontiers in Probabilistic Inference workshop at International Conference on Machine Learining (ICML)*, 2025.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Ping He, Om Khangaonkar, Hamed Pirsiavash, Yikun Bai, and Soheil Kolouri. Sinkhorn-drifting generative models. *arXiv preprint arXiv:2603.12366*, 2026.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Neural Information Processing Systems (NeurIPS)*, 2020.

Matthew D Hoffman. Learning deep latent Gaussian models with Markov chain Monte Carlo. *International Conference on Machine Learning (ICML)*, 2017.

Lars Holdijk, Yuanqi Du, Ferry Hooft, Priyank Jaini, Berend Ensing, and Max Welling. Stochastic optimal control for collective variable free sampling of molecular transition paths. *Neural Information Processing Systems (NeurIPS)*, 2023.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. *International Conference on Machine Learning (ICML)*, 2022.

Yiwen Huang, Aaron Gokaslan, Volodymyr Kuleshov, and James Tompkin. The GAN is dead; long live the GAN! A modern GAN baseline. *Neural Information Processing Systems (NeurIPS)*, 2024.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the Fokker–Planck equation. *SIAM journal on mathematical analysis*, 1998.

Adhithyan Kalaivanan, Zheng Zhao, Jens Sjölund, and Fredrik Lindsten. ESS-Flow: Training-free guidance of flow-based models as inference in source space. *arXiv preprint arXiv:2510.05849*, 2025.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations (ICLR)*, 2014.

Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo Rohde. Generalized sliced Wasserstein distances. *Neural Information Processing Systems (NeurIPS)*, 2019.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *International Conference on Learning Representations (ICLR)*, 2021.

Salem Lahlou, Tristan Deleu, Pablo Lemos, Dinghuai Zhang, Alexandra Volokhova, Alex Hernández-Garcıa, Léna Néhale Ezzine, Yoshua Bengio, and Nikolay Malkin. A theory of continuous generative flow networks. *International Conference on Machine Learning (ICML)*, 2023.

Benedict Leimkuhler and Charles Matthews. Rational construction of stochastic numerical methods for molecular sampling. *Applied Mathematics Research eXpress*, 2013a.

Benedict Leimkuhler and Charles Matthews. Robust and efficient configurational molecular sampling via Langevin dynamics. *The Journal of Chemical Physics*, 138(17), 2013b.

Benedict Leimkuhler and Charles Matthews. *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*. Springer, 2015.

Yingzhen Li, Richard E Turner, and Qiang Liu. Approximate inference with amortised MCMC. *arXiv preprint arXiv:1702.08343*, 2017.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. *Neural Information Processing Systems (NeurIPS)*, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*, 2019.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 1953.

Laurence Illing Midgley, Vincent Stimper, Gregor NC Simm, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Flow annealed importance sampling bootstrap. *International Conference on Learning Representations (ICLR)*, 2023.

Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

Petr Mokrov, Alexander Korotin, Lingxiao Li, Aude Genevay, Justin M Solomon, and Evgeny Burnaev. Large-scale Wasserstein gradient flows. *Neural Information Processing Systems (NeurIPS)*, 2021.

Kimia Nadjahi, Alain Durmus, Lénaïc Chizat, Soheil Kolouri, Shahin Shahrampour, and Umut Simsekli. Statistical and topological properties of sliced probability divergences. *Neural Information Processing Systems (NeurIPS)*, 2020.

Radford M Neal. Slice sampling. *The Annals of Statistics*, 2003.

Radford M Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*, 2011.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 2019.

Kiyoung Om, Kyuil Sim, Taeyoung Yun, Hyeongyu Kang, and Jinkyoo Park. Posterior inference in latent space for scalable constrained black-box optimization. *arXiv preprint arXiv:2507.00480*, 2025.

Felix Otto and Cédric Villani. Generalization of an inequality by Talagrand and links with the logarithmic Sobolev inequality. *Journal of Functional Analysis*, 2000.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. *International Conference on Machine Learning (ICML)*, 2015.

Lorenz Richter, Ayman Boustati, Nikolas Nüsken, Francisco J. R. Ruiz, and Ömer Deniz Akyildiz. VarGrad: A low-variance gradient estimator for variational inference. *Neural Information Processing Systems (NeurIPS)*, 2020.

Lorenz Richter, Julius Berner, and Guan-Horng Liu. Improved sampling via learned diffusions. *International Conference on Learning Representations (ICLR)*, 2024.

Gareth O Roberts and Richard L Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. 1996.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

Peter J Rossky, Jimmie D. Doll, and Harold L. Friedman. Brownian dynamics as smart Monte Carlo simulation. *The Journal of Chemical Physics*, 69:4628–4633, 1978.

Francisco Ruiz and Michalis Titsias. A contrastive divergence for combining variational inference and MCMC. *International Conference on Machine Learning (ICML)*, 2019.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Neural Information Processing Systems (NeurIPS)*, 2024.

Sergey Samsonov, Evgeny Lagutin, Marylou Gabrié, Alain Durmus, Alexey Naumov, and Eric Moulines. Local-global MCMC kernels: the best of both worlds. *Neural Information Processing Systems (NeurIPS)*, 2022.

Filippo Santambrogio. Optimal transport for applied mathematicians: Calculus of variations, PDEs, and modeling, volume 87 of progress in nonlinear differential equations and their applications. 2015.

Filippo Santambrogio. {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of Mathematical Sciences*, 7(1):87–154, 2017.

Marcin Sendera, Minsu Kim, Sarthak Mittal, Pablo Lemos, Luca Scimeca, Jarrid Rector-Brooks, Alexandre Adam, Yoshua Bengio, and Esmeralda S. Whitammer. Improved off-policy training of diffusion samplers. *Neural Information Processing Systems (NeurIPS)*, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *International Conference on Machine Learning (ICML)*, 2015.

Jiaming Song, Shengjia Zhao, and Stefano Ermon. A-NICE-MC: Adversarial training for MCMC. *Neural Information Processing Systems (NeurIPS)*, 2017.

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

Gilbert Strang. On the construction and comparison of difference schemes. *SIAM journal on numerical analysis*, 5(3):506–517, 1968.

Kirill Tamogashev and Esmeralda S. Whitammer. Data-to-energy stochastic dynamics. *International Conference on Learning Representations (ICLR)*, 2026.

Zhiwei Tang, Jiangweizhi Peng, Jiasheng Tang, Mingyi Hong, Fan Wang, and Tsung-Hui Chang. Inference-time alignment of diffusion models with direct noise optimization. *International Conference on Machine Learning (ICML)*, 2025.

Chenyang Tian, Bharath K Sriperumbudur, Arthur Gretton, and Zonghao Chen. Sobolev regularized MMD gradient flow. *arXiv preprint arXiv:2605.11884*, 2026.

Michalis K Titsias. Learning model reparametrizations: Implicit variational inference by fitting MCMC distributions. *arXiv preprint arXiv:1708.01529*, 2017.

Francisco Vargas, Will Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. *International Conference on Learning Representations (ICLR)*, 2023.

Santosh Vempala and Andre Wibisono. Rapid convergence of the unadjusted Langevin algorithm: Isoperimetry suffices. *Neural Information Processing Systems (NeurIPS)*, 2019.

Siddarth Venkatraman, Mohsin Hasan, Minsu Kim, Luca Scimeca, Marcin Sendera, Yoshua Bengio, Glen Berseth, and Nikolay Malkin. Outsourced diffusion sampling: Efficient posterior inference in latent spaces of generative models. *International Conference on Machine Learning (ICML)*, 2025.

Max-K. von Renesse and Karl-Theodor Sturm. Transport inequalities, gradient estimates, entropy and Ricci curvature. *Communications on Pure and Applied Mathematics*, 2005.

Congye Wang, Wilson Chen, Heishiro Kanagawa, and Chris Oates. Reinforcement learning for adaptive MCMC. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2025.

Zifan Wang, Alice Harting, Matthieu Barreau, Michael Zavlanos, and Karl H Johansson. Source-guided flow matching. *International Conference on Learning Representations (ICLR)*, 2026.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. *International Conference on Machine Learning (ICML)*, 2011.

Peter Wirnsberger, Andrew J Ballard, George Papamakarios, Stuart Abercrombie, Sébastien Racanière, Alexander Pritzel, Danilo Jimenez Rezende, and Charles Blundell. Targeted free energy estimation via learned mappings. *The Journal of Chemical Physics*, 2020.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. *Neural Information Processing Systems (NeurIPS)*, 2023.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: a stochastic control approach for sampling. *International Conference on Learning Representations (ICLR)*, 2022.

A Additional discussion of the algorithm

A.1 Vector-field-based loss

It is possible to an consider alternative training procedure for OSIS, which is similar to the one used for drifting models (Deng et al., 2026; Gretton et al., 2026). Gradient descent on the parameters θ along the objective in (2) approximates the Wasserstein gradient descent of the distribution ρ along the *Wasserstein gradients of the divergence* $\mathcal{D}$ (see §3). The theory of Wasserstein geometry (Ambrosio et al., 2005) tells us that the evolution of a distribution can be identified with the evolution of particles that follow this distribution along a gradient field. In this case, the evolution of particles follows a vector field V such that $V = -\nabla \frac{\delta \mathcal{D}}{\delta \rho_\theta}$, with the term on the right being the gradient of the first variation of $\mathcal{D}$. This observation allows us to use an alternative training objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{z\sim q}\left[\|G_\theta(z) - \mathrm{sg}\left[G_\theta(z) + \eta V(G_\theta(z))\right]\|^2\right], \tag{12}$$

where now V is estimated empirically using samples from ρ_θ. The algorithm corresponding to this objective is presented in Algorithm 2.

A.2 Full algorithm

Algorithms 1 and 2 describe the idealised version of OSIS. However, as discussed in §5, there are many design choices that can help improve performance of the algorithm in practice. Algorithms 3 and 4 present the complete version of OSIS and include tempering, approximation of a Markov chain and the divergence with their empirical analogues, as well as the use of replay buffer and inner iterations. We note that this work focuses on the variant of OSIS described in Algorithms 1 and 3. Pseudocode in Algorithms 2 and 4 is provided for completeness.

Algorithm 2 OSIS using a vector field V

Require: Prior distributon q, π-invariant kernel K_π, number of chain steps M, generator G_θ, vector field $V_{(\cdot,\cdot)}$, step size η.

while not converged **do**
  $\rho_\theta \leftarrow G_{\theta\#}q$
  ▷ *Run a chain for time h*
  $\rho_h \leftarrow K^h \rho_{\mathrm{sg}[\theta]}$
  ▷ *Compute loss using V*
  $\mathcal{L}(\theta) \leftarrow \mathbb{E}_{z\sim q(z), x=G_\theta(z)} \| G_\theta(z) - \mathrm{sg}\left[x + V_{\rho_{\mathrm{sg}[\theta]},\rho_h}(x)\right] \|^2$
  Gradient update on θ with $\nabla_\theta \mathcal{L}$
end while
return G_θ

Algorithm 3 Full OSIS algorithm with $\hat{\mathcal{D}}$

Require: Prior distribution q, generator G_θ, π-invariant kernel K, empirical divergence $\hat{\mathcal{D}}$, number of outer steps N, number of inner steps M, number of chain steps N_{MCMC}, buffer size B, batch size $\tilde{B}$,

for `outer` $= 1, \ldots, N$ **do**
  $\{z_i\}_{i=1}^B \sim \mathcal{N}(0, I)$, $x_i \leftarrow G_{\mathrm{sg}[\theta]}(z_i)$
  ▷ *Run a chain for N_{MCMC} steps*
  ▷ *with inverse temperature β*
  $\beta \leftarrow$ get_inverse_temp(`outer`)
  $y_i \leftarrow$ run_mcmc$(x_i, K, N_{\mathrm{MCMC}}, \beta)$
  ▷ *Instantiate buffer and save samples*
  Instantiate buffer $\mathcal{B}$
  with $\left(\{z_i\}_{i=1}^B, \{y_i\}_{i=1}^B\right)$
  ▷ *Run several inner iterations*
  ▷ *for computational stability*
  for `inner` $= 1, \ldots, M$ **do**
    if `resample` is **True then**
      $\left(\{z_i\}_{i=1}^{\tilde{B}}, \{y_i\}_{i=1}^{\tilde{B}}\right) \sim \mathcal{B}$
    else
      Get $\left(\{z_i\}_{i=1}^{\tilde{B}}, \{y_i\}_{i=1}^{\tilde{B}}\right)$ from $\mathcal{B}$
    end if
    $x_i^\theta \leftarrow G_\theta(z_i)$
    $\hat{p}_x^\theta \leftarrow \sum_{j=1}^{\tilde{B}} \delta_{x_j^\theta}$, $\hat{p}_y \leftarrow \sum_{j=1}^{\tilde{B}} \delta_{y_j}$
    $\mathcal{L}(\theta) \leftarrow \hat{\mathcal{D}}(\hat{p}_x^\theta, \hat{p}_y)$
    Gradient update on θ with $\nabla_\theta \mathcal{L}$
  end for
end for
return G_θ

Algorithm 4 Full OSIS algorithm with V

Require: Prior distribution q, generator G_θ, π-invariant kernel K, vector field $V_{(\cdot,\cdot)}$, number of outer steps N, number of inner steps M, number of chain steps N_{MCMC}, buffer size B, batch size $\tilde{B}$, step size η.

for `outer` $= 1, \ldots, N$ **do**
  $\{z_i\}_{i=1}^B \sim \mathcal{N}(0, I)$, $x_i \leftarrow G_\theta(z_i)$
  ▷ *Run a chain for N_{MCMC} steps*
  ▷ *with inverse temperature β*
  $\beta \leftarrow$ get_inverse_temp(`outer`)
  $y_i \leftarrow$ run_mcmc$(\mathrm{sg}\,[x_i], K, N_{\mathrm{MCMC}}, \beta)$
  ▷ *Instantiate buffer and save samples*
  Instantiate buffer $\mathcal{B}$ with $\left(\{z_i\}_{i=1}^B, \{y_i\}_{i=1}^B\right)$
  ▷ *Run several inner iterations*
  ▷ *for computational stability*
  for `inner` $= 1, \ldots, M$ **do**
    if `resample` is **True then**
      $\left(\{z_i\}_{i=1}^{\tilde{B}}, \{y_i\}_{i=1}^{\tilde{B}}\right) \sim \mathcal{B}$
    else
      Get $\left(\{z_i\}_{i=1}^{\tilde{B}}, \{y_i\}_{i=1}^{\tilde{B}}\right)$ from $\mathcal{B}$
    end if
    $x_i^\theta \leftarrow G_\theta(z_i)$
    ▷ $z \sim q$, $x = G_\theta(z)$
    $\mathcal{L}(\theta) \leftarrow$
      $\mathbb{E}_{z,x} \| G_\theta(z) - \mathrm{sg}\left[x + \eta V_{\rho_{\mathrm{sg}[\theta]},\rho_h}(x)\right] \|^2$
    Gradient update on θ with $\nabla_\theta \mathcal{L}$
  end for
end for
return G_θ

B Extended theoretical analysis

B.1 Additional notation and temperature scaling

KL divergence and Fisher information. For distribution $\rho, \mu \in \mathcal{P}_2(\mathbb{R}^d)$, we define KL divergence and Fisher information as follows:

$$\mathrm{KL}\left(\rho \,\|\, \mu\right) \coloneqq \int \rho(x) \log \frac{\rho(x)}{\mu(x)} \,\mathrm{d}x, \qquad (13) \qquad \mathrm{FI}\left(\rho \,\|\, \mu\right) \coloneqq \int \left|\nabla \log \frac{\rho(x)}{\mu(x)}\right|^2 \rho(x) \,\mathrm{d}x. \qquad (14)$$

We assume that for all cases we consider both KL divergence and Fisher information are finite.

2-Wasserstein distance. For distributions $\rho, \mu \in \mathcal{P}_2(\mathbb{R}^d)$, we define 2-Wasserstein distance as

$$\mathcal{W}_2^2(\rho, \mu) = \inf \left\{ \int_{\mathbb{R}^d \times \mathbb{R}^d} |x - y|^2 \, \gamma(x, y) \mathrm{d}x \mathrm{d}y \quad \text{s.t.} \quad \gamma \in \Gamma(\rho, \mu) \right\}, \qquad (15)$$

$\Gamma(\rho, \mu)$ is the coupling of probability measures such that its marginals are ρ and μ. (15) defines the Kantorovich version of OT. For measures in $\mathcal{P}_2(\mathbb{R}^d)$ the minimal coupling always exists and is unique; moreover, (15) satisfies triangular inequality, meaning that 2-Wasserstein is a distance metric in $\mathcal{P}_2(\mathbb{R}^d)$. Moreover, for $\mathcal{P}_2(\mathbb{R}^d)$ the optimal coupling has the form of $\gamma^* = (\mathrm{Id}, T)_{\#}\rho$, where T is the Monge OT map. This allows us to introduce one more (in this case equivalent) definition of 2-Wasserstein distance:

$$\mathcal{W}_2^2(\rho, \mu) = \inf \left\{ \int |T(x) - x|^2 \, \rho(x) \mathrm{d}x \quad \text{s.t.} \quad T_{\#}\rho = \mu \right\}. \qquad (16)$$

Another useful definition of 2-Wasserstein distance, which we use to prove Lemma B.3 is due to Benamou-Brenier (Ambrosio et al., 2005, Equation 8.0.3) and employs the dynamic formulation of optimal transport:

$$\mathcal{W}_2^2(\rho, \mu) = \inf \left\{ \int_0^1 \int |v_t|^2 \rho_t(x) \mathrm{d}x \mathrm{d}t \quad \text{s.t.} \quad \frac{\partial}{\partial t}\rho_t + \nabla \cdot (\rho_t \, v_t) = 0, \, \rho_0 = \rho, \, \rho_1 = \mu \right\}. \qquad (17)$$

B.2 Proofs

We first recall some standard definitions and results in Wasserstein geometry, see, *e.g.*, Santambrogio (2015).

Definition B.1 (Geodesic convexity)**.** *A functional $\mathcal{J}(\rho) : \mathcal{P}_2(\mathbb{R}^d) \to \mathbb{R}$ is said to be*

- ***geodesically convex** if, when ρ_t is the geodesic between ρ_0 and ρ_1, $\rho_0, \rho_1 \in \mathcal{P}_2(\mathbb{R}^d)$,*

$$\forall t \in [0, 1] \quad \mathcal{J}(\rho_t) \le t\mathcal{J}(\rho_1) + (1 - t)\mathcal{J}(\rho_0); \qquad (18)$$

- *λ-**strongly geodesically convex** if, when ρ_t is the geodesic between ρ_0 and ρ_1, $\rho_0, \rho_1 \in \mathcal{P}_2(\mathbb{R}^d)$,*

$$\forall t \in [0, 1] \quad \mathcal{J}(\rho_t) \le t\mathcal{J}(\rho_1) + (1 - t)\mathcal{J}(\rho_0) - \frac{\lambda}{2} t(1 - t)\mathcal{W}_2^2(\rho_0, \rho_1). \qquad (19)$$

Lemma B.2 (KL convexity along a Wasserstein geodesic)**.** *Let $\rho_0, \rho_1 \in \mathcal{P}_2(\mathbb{R}^d)$, and let ρ_t be the geodesic between ρ_0 and ρ_1 in $\mathcal{P}_2(\mathbb{R}^d)$. With $\pi(x) = Z^{-1}e^{-\mathcal{E}(x)}$, $\mathrm{KL}\left(\rho_t \,\|\, \pi\right)$ is*

- ***geodesically convex** if π is log-concave;*
- *λ-**strongly geodesically convex** if π is λ-strongly log-concave.*

Proof of Lemma B.2. We prove the result for λ-strong convexity; convexity can be recovered by setting $\lambda = 0$. Recall that $\rho_t = (T_t)_{\#}\rho_0$, where $T_t(x) = tT + (1 - t)\,\mathrm{Id}$ and T is the Monge map from

ρ_0 to ρ_1. First, we rewrite $\mathrm{KL}(\rho_t \,\|\, \pi)$ as follows:

$$\mathrm{KL}(\rho_t \,\|\, \pi) = \int \rho_t(x) \log \frac{\rho_t(x)}{\pi(x)} \mathrm{d}x = \int \rho_0(x) \log \frac{\rho_0(x)}{\pi(T_t(x)) \det(\nabla T_t)} \mathrm{d}x \tag{20}$$

$$= \int \mathcal{E}(T_t(x)) \rho_0(x)\, \mathrm{d}x - \int \log\det(\nabla T_t) \rho_0(x)\, \mathrm{d}x + \text{const.} \tag{21}$$

Next, we show the first term is λ-strongly convex and the second term is convex. Since π is λ-strongly log-concave, for any x we have

$$\mathcal{E}(T_t(x)) = \mathcal{E}(tT(x) + (1-t)x)t \le \mathcal{E}(T(x)) + (1-t)\mathcal{E}(x) - \frac{\lambda}{2} t(1-t)|T(x) - x|^2. \tag{22}$$

Integrating with $\rho_0(x)$, we obtain

$$\int \mathcal{E}(T_t(x))\rho_0(x)\mathrm{d}x \le \int t\mathcal{E}(T(x))\rho_0(x)\mathrm{d}x + (1-t)\int \mathcal{E}(x)\rho_0(x)\mathrm{d}x - \frac{\lambda}{2} t(1-t)\mathcal{W}_2^2(\rho_0, \rho_1), \tag{23}$$

showing λ-strong convexity of the first term. For the second term, we have

$$-\log\det(\nabla T_t) = -\log\det(t\nabla T(x) + (1-t)\,\mathrm{Id}) = -\sum_i \log((1-t) + t\sigma_i) \le -t\log\det(\nabla T), \tag{24}$$

where σ_i are the eigenvalues of ∇T. Because T is the gradient of a convex function (Santambrogio, 2015) $t\log\det(\nabla T) > 0$, integrating over ρ_0, we get

$$-\int \log\det(\nabla T_t)\rho_0(x)\, \mathrm{d}x \le -t\int \log\det(\nabla T)\rho_0(x)\, \mathrm{d}x, \tag{25}$$

which is exactly convexity of the second summand, noting that $T_0 = \mathrm{Id}$ (so $\log\det(\nabla T_0) = 0$) and $T_1 = T$.

This proves λ-strong convexity of KL along ρ_t. The case of geodesic convexity can be recovered by setting $\lambda = 0$. □

Lemma B.3. *Let $\rho_0 \in \mathcal{P}_2(\mathbb{R}^d)$ satisfy $\mathrm{KL}(\rho_0 \,\|\, \pi) < \infty$, and define $\rho_h = K^h \rho_0$, Then, for any $h > 0$,*

$$\mathcal{W}_2^2(\rho_0, \rho_h) \le h\,\mathrm{KL}(\rho_0 \,\|\, \pi)\,. \tag{26}$$

Proof of Lemma B.3. We begin the proof by recalling the Benamou-Brenier formula for $\mathcal{W}_2(\rho_0, \rho_h)$ (see (17)). For the velocity $v_t = \nabla \log \frac{\pi}{\rho_t}$, which is associated with the Langevin diffusion, we have

$$\mathcal{W}_2^2(\rho_0, \rho_h) \le h\int_0^h \int |v_t|^2 \rho_t(x)\mathrm{d}x\mathrm{d}t = h^2 \int_0^1 \int |v_{hs}|^2 \rho_{hs}(x)\mathrm{d}x\mathrm{d}s \tag{27}$$

$$= h^2 \int_0^1 \mathrm{FI}(\rho_{hs} \,\|\, \pi)\, ds = h(\mathrm{KL}(\rho_0 \,\|\, \pi) - \mathrm{KL}(\rho_h \,\|\, \pi)) \le h\,\mathrm{KL}(\rho_0 \,\|\, \pi)\,, \tag{28}$$

where the first inequality follows from the Langevin diffusion path having a greater dynamic transport cost than the geodesic, the second equality is due to rescaling $s = t/h$, and the last equality is due to the fact that $\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{KL}(\rho_t \,\|\, \pi) = -\mathrm{FI}(\rho_t \,\|\, \pi)$, (Gretton et al., 2026, §2.2), (Vempala & Wibisono, 2019, Lemma 2). □

Theorem 4.1 (Contraction of one step of OSIS under λ-SLC π)**.** *Let π be λ-SLC. Let $\rho_k, \rho_{h_k}, T^{h_k}, T_{\eta_k}^{h_k}, \rho_{k+1}$ be as in §4.1. For any $h_k > 0$, $\eta_k \in (0, 1]$, the following bounds hold:*

$$\mathcal{W}_2(\rho_{k+1}, \pi) \le \left((1-\eta_k)(1 + e^{-\lambda h_k}) + e^{-\lambda h_k}\right)\mathcal{W}_2(\rho_k, \pi); \tag{8}$$

$$\mathrm{KL}(\rho_{k+1} \,\|\, \pi) \le (1 - \eta_k(1 - e^{-2\lambda h_k}))\,\mathrm{KL}(\rho_k \,\|\, \pi)\,. \tag{9}$$

Proof of Theorem 4.1. We start by recalling the Langevin convergence bound for λ-SLC distributions (von Renesse & Sturm, 2005, Theorem 1.1(iii)):

$$\mathcal{W}_2(\rho_{h_k}, \pi) \le e^{-\lambda h_k}\mathcal{W}_2(\rho_k, \pi), \tag{29}$$

which, together with the triangular inequality, provides the following bound for $\mathcal{W}_2(\rho_{k+1}, \pi)$:

$$\mathcal{W}_2(\rho_{k+1}, \pi) \le \mathcal{W}_2(\rho_{k+1}, \rho_{h_k}) + e^{-\lambda h_k}\mathcal{W}_2(\rho_k, \pi).$$

Next, we introduce the bound for $W_2(\rho_{k+1}, \rho_{h_k})$

$$\begin{aligned}\mathcal{W}_2(\rho_{k+1}, \rho_{h_k}) &= (1-\eta_k)\mathcal{W}_2(\rho_k, \rho_{h_k}) \le (1-\eta_k)\Big(\mathcal{W}_2(\rho_k, \pi) + \mathcal{W}_2(\rho_{h_k}, \pi)\Big) \\ &\le (1-\eta_k)\Big(\mathcal{W}_2(\rho_k, \pi) + e^{-\lambda h_k}\mathcal{W}_2(\rho_k, \pi)\Big) = (1-\eta_k)(1+e^{-\lambda h_k})\mathcal{W}_2(\rho_k, \pi).\end{aligned}$$

This gives us the combined bound:

$$\mathcal{W}_2(\rho_{k+1}, \pi) \le \underbrace{\Big(\underbrace{(1-\eta_k)(1+e^{-\lambda h_k})}_{\substack{\text{penalty for the} \\ \text{gradient descent}}} + \underbrace{e^{-\lambda h_k}}_{\substack{\text{convergence} \\ \text{of exact Langevin}}}\Big)}_{=C(h_k, \eta_k)} \mathcal{W}_2(\rho_k, \pi).$$

Now we establish conditions for $C(h_k, \eta_k) < 1$:

$$\begin{aligned}&C(h_k, \eta_k) = 1 + 2e^{-\lambda h_k} - \eta_k(1+e^{-\lambda h_k}) < 1, \\ &2e^{-\lambda h_k} < \eta_k(1+e^{-\lambda h_k}), \quad 2/(1+e^{\lambda h_k}) < \eta_k.\end{aligned}$$

So we get $\eta_k > 2\sigma(-\lambda h_k)$, where $\sigma(s)$ is the sigmoid function.

We now prove the bound on KL divergence under the same assumptions. We use the strong convexity of KL along geodesics (Lemma B.2) to obtain the following inequality:

$$\mathrm{KL}\left(\rho_{k+1} \,\|\, \pi\right) \le (1-\eta_k)\,\mathrm{KL}\left(\rho_k \,\|\, \pi\right) + \eta_k\,\mathrm{KL}\left(\rho_{h_k} \,\|\, \pi\right) - \frac{\lambda}{2}\eta_k(1-\eta_k)\mathcal{W}_2^2(\rho_k, \rho_{h_k}). \tag{30}$$

We recall that $\rho_{h_k} = K^{h_k}\rho_k$. If ρ_t is a continuous curve in $\mathcal{P}_2(\mathbb{R}^d)$ induced by Langevin diffusion, using the Bakry-Émery criterion (Bakry & Émery, 1985), together with the fact that $\frac{\mathrm{d}}{\mathrm{d}t}\,\mathrm{KL}\left(\rho_t \,\|\, \pi\right) = -\,\mathrm{FI}\left(\rho_t \,\|\, \pi\right)$, (Gretton et al., 2026, §2.2), Vempala & Wibisono (2019, Lemma 2) gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\,\mathrm{KL}\left(\rho_t \,\|\, \pi\right) = -\,\mathrm{FI}\left(\rho_t \,\|\, \pi\right) \le -2\lambda\,\mathrm{KL}\left(\rho_t \,\|\, \pi\right), \tag{31}$$

which consequently implies geometric convergence for KL:

$$\mathrm{KL}\left(\rho_{h_k} \,\|\, \pi\right) \le e^{-2\lambda h_k}\,\mathrm{KL}\left(\rho_k \,\|\, \pi\right). \tag{32}$$

Finally, combining Equation (32) with (30) gives:

$$\begin{aligned}\mathrm{KL}\left(\rho_{k+1} \,\|\, \pi\right) &\le (1-\eta_k)\,\mathrm{KL}\left(\rho_k \,\|\, \pi\right) + \eta_k\,\mathrm{KL}\left(\rho_{h_k} \,\|\, \pi\right) - \frac{\lambda}{2}\eta_k(1-\eta_k)\mathcal{W}_2^2(\rho_k, \rho_{h_k}) \\ &\le (1-\eta_k(1-e^{-2\lambda h_k}))\,\mathrm{KL}\left(\rho_k \,\|\, \pi\right) - \frac{\lambda}{2}\eta_k(1-\eta_k)\mathcal{W}_2^2(\rho_k, \rho_{h_k}) \\ &\le \underbrace{\Big(1-\eta_k(1-e^{-2\lambda h_k})\Big)}_{C(\eta_k, h_k)}\,\mathrm{KL}\left(\rho_k \,\|\, \pi\right),\end{aligned} \tag{33}$$

which proves the final bound for KL with $C(\eta_k, h_k) = 1 - \eta_k(1-e^{-2\lambda h_k})$. □

Theorem 4.2 (Convergence of OSIS under λ-SLC π)**.** *Under the assumptions of Theorem 4.1, let $(\rho_k)_{k\ge 0}$ be the sequence of probability distributions produced by the* OSIS *iterations. Assuming $h_k > 0$ and $\eta_k \in (0, 1]$ for all $k \ge 0$, the following bounds hold for any $N \ge 1$:*

$$\mathrm{KL}\left(\rho_N \,\|\, \pi\right) \le e^{-\sum_{k=0}^{N-1}\eta_k(1-e^{-2\lambda h_k})}\,\mathrm{KL}\left(\rho_0 \,\|\, \pi\right). \tag{10}$$

Moreover, if $\bar{h} = \inf_{k\ge 0} h_k > 0$, $\bar{\eta} = \inf_{k\ge 0}\eta_k > 0$, and $\bar{\eta} > 2/(1+\exp(\lambda\bar{h}))$, then:

$$\mathcal{W}_2(\rho_N, \pi) \le \left((1-\bar{\eta})(1+e^{-\lambda\bar{h}}) + e^{-\lambda\bar{h}}\right)^N \mathcal{W}_2(\rho_0, \pi); \tag{11}$$

Proof of Theorem 4.2. By iteratively applying the bound from Theorem 4.1 N times and using $0 < 1 - x < e^{-x}$ for $x < 1$, we get:

$$\mathrm{KL}\left(\rho_N \,\|\, \pi\right) \leq \prod_{k=0}^{N-1} \left(1 - \eta_k(1 - e^{-2\lambda h_k})\right) \mathrm{KL}\left(\rho_0 \,\|\, \pi\right) \leq \exp\Big(-\sum_{k=0}^{N-1} \eta_k(1 - e^{-2\lambda h_k})\Big) \mathcal{W}_2^2(\rho_0, \pi).$$

Hence $\mathrm{KL}\left(\rho_k \,\|\, \pi\right) \to 0$ whenever $\lim_{N\to\infty} \sum_{k=0}^{N-1} \eta_k(1 - e^{-2\lambda h_k}) \to \infty$.

To prove the bound for $\mathcal{W}_2$, we observe that $\forall k \geq 0 \quad C(\bar{\eta}, \bar{h}) < C(\eta_k, h_k)$. By recursively applying the bound from (8) we get:

$$\mathcal{W}_2(\rho_N, \pi) \leq \left((1 - \bar{\eta})(1 + e^{-\lambda \bar{h}}) + e^{-\lambda \bar{h}}\right)^N \mathcal{W}_2(\rho_0, \pi). \tag{34}$$

□

Theorem B.4 (Contraction of one step of OSIS under π satisfying logarithmic Sobolev inequality)**.** *Suppose that $\mathcal{E} \in C^2(\mathbb{R}^d)$ and $\nabla^2(\mathcal{E}) \succeq -\kappa I$ for $\kappa \geq 0$ and that $\pi \propto e^{-\mathcal{E}}$ satisfies the logarithmic Sobolev inequality with parameter $\alpha > 0$ (α-LSI), that is:*

$$\forall \rho \in \mathcal{P}(\mathbb{R}^d) \qquad 2\alpha \,\mathrm{KL}\left(\rho \,\|\, \pi\right) \leq \mathrm{FI}\left(\rho \,\|\, \pi\right). \tag{35}$$

With the same notations as in Theorem 4.1, the following bound holds:

$$\mathrm{KL}\left(\rho_{k+1} \,\|\, \pi\right) \leq \mathrm{KL}\left(\rho_k \,\|\, \pi\right) - \eta_k \left[1 - \frac{\kappa h_k}{2}(1 - \eta_k)\right] \left[\mathrm{KL}\left(\rho_k \,\|\, \pi\right) - \mathrm{KL}\left(\rho_{h_k} \,\|\, \pi\right)\right]. \tag{36}$$

Further, for $h > 0$, $\eta \in (0, 1]$, define

$$C_{\alpha,\kappa}(\eta, h) \coloneqq 1 - \eta \left[1 - \frac{\kappa h}{2}(1 - \eta)\right](1 - e^{-2\alpha h}). \tag{37}$$

If $\kappa h_k(1 - \eta_k) \leq 2$, then

$$\mathrm{KL}\left(\rho_{k+1} \,\|\, \pi\right) \leq C_{\alpha,\kappa}(\eta_k, h_k) \,\mathrm{KL}\left(\rho_k \,\|\, \pi\right). \tag{38}$$

Proof of Theorem B.4. Apply Lemma B.2 to the Wasserstein geodesic from ρ_k to $\rho_{h_k} = K^{h_k}\rho_k$, with curvature parameter $-\kappa$ and interpolation parameter η_k. This gives

$$\begin{aligned}\mathrm{KL}\left(\rho_{k+1} \,\|\, \pi\right) \leq (1 - \eta_k)\,\mathrm{KL}\left(\rho_k \,\|\, \pi\right) + \eta_k \,\mathrm{KL}\left(\rho_{h_k} \,\|\, \pi\right) \\ + \frac{\kappa}{2}\eta_k(1 - \eta_k)\mathcal{W}_2^2(\rho_k, \rho_{h_k}).\end{aligned} \tag{39}$$

The last term is nonnegative because $\kappa \geq 0$ and $\eta_k \in (0, 1]$.

Rearranging the first two entropy terms gives

$$\begin{aligned}\mathrm{KL}\left(\rho_{k+1} \,\|\, \pi\right) \leq \mathrm{KL}\left(\rho_k \,\|\, \pi\right) - \eta_k \left[\mathrm{KL}\left(\rho_k \,\|\, \pi\right) - \mathrm{KL}\left(\rho_{h_k} \,\|\, \pi\right)\right] \\ + \frac{\kappa}{2}\eta_k(1 - \eta_k)\mathcal{W}_2^2(\rho_k, \rho_{h_k}).\end{aligned} \tag{40}$$

Using the proof of Lemma B.3,

$$\mathcal{W}_2^2(\rho_k, \rho_{h_k}) \leq h_k \left[\mathrm{KL}\left(\rho_k \,\|\, \pi\right) - \mathrm{KL}\left(\rho_{h_k} \,\|\, \pi\right)\right].$$

The multiplier $\frac{\kappa}{2}\eta_k(1 - \eta_k)$ is nonnegative, so substituting this upper bound into (40) we obtain

$$\begin{aligned}\mathrm{KL}\left(\rho_{k+1} \,\|\, \pi\right) \leq \mathrm{KL}\left(\rho_k \,\|\, \pi\right) \\ - \eta_k \left[1 - \frac{\kappa h_k}{2}(1 - \eta_k)\right] \left[\mathrm{KL}\left(\rho_k \,\|\, \pi\right) - \mathrm{KL}\left(\rho_{h_k} \,\|\, \pi\right)\right].\end{aligned} \tag{41}$$

Exact Langevin performs descent on relative entropy, so $\mathrm{KL}\left(\rho_k \,\|\, \pi\right) - \mathrm{KL}\left(\rho_{h_k} \,\|\, \pi\right) \geq 0$. We conclude that, if $\kappa h_k(1 - \eta_k) \leq 2$ then

$$1 - \frac{\kappa h_k}{2}(1 - \eta_k) \geq 0,$$

and so the subtracted term is nonnegative, proving entropy monotonicity.

For almost every $s > 0$, the entropy dissipation identity and α-LSI bound give

$$\frac{\mathrm{d}}{\mathrm{d}s} \mathrm{KL}\left(K^s \rho_k \,\|\, \pi\right) = -\,\mathrm{FI}\left(K^s \rho_k \,\|\, \pi\right) \le -2\alpha\, \mathrm{KL}\left(K^s \rho_k \,\|\, \pi\right). \tag{42}$$

Integrating as in the proof of Lemma B.3,

$$\mathrm{KL}\left(\rho_{h_k} \,\|\, \pi\right) \le e^{-2\alpha h_k}\, \mathrm{KL}\left(\rho_k \,\|\, \pi\right). \tag{43}$$

We substitute this upper-bound in (41) noting that the step condition makes $\eta_k[1 - \kappa h_k(1-\eta_k)/2]$ nonnegative and obtain

$$\mathrm{KL}\left(\rho_{k+1} \,\|\, \pi\right) \le C_{\alpha,\kappa}(\eta_k, h_k)\, \mathrm{KL}\left(\rho_k \,\|\, \pi\right). \tag{44}$$

□

Theorem B.5 (Convergence under LSI and bounded negative curvature)**.** *Suppose that $\mathcal{E} \in C^2(\mathbb{R}^d)$ and $\nabla^2(\mathcal{E}) \succeq -\kappa I$ for $\kappa \ge 0$ and that $\pi \propto e^{-\mathcal{E}}$ satisfies α-LSI. With the same notations as in Theorem 4.1, let $(\rho_k)_{k\ge 0}$ be the sequence of probability distributions produced by* OSIS *interations, define*

$$C_{\alpha,\kappa}(\eta, h) = 1 - \eta\left[1 - \frac{\kappa h}{2}(1-\eta)\right](1 - e^{-2\alpha h}). \tag{45}$$

Assuming $h_k > 0$, $\eta_k \in (0, 1]$ for all $k \ge 0$, the following bounds hold for any $N \ge 1$:

$$\mathrm{KL}\left(\rho_N \,\|\, \pi\right) \le \left(\prod_{k=0}^{N-1} C_{\alpha,\kappa}(\eta_k, h_k)\right) \mathrm{KL}\left(\rho_0 \,\|\, \pi\right) \tag{46}$$

$$\le \exp\left(-\sum_{k=0}^{N-1} [1 - C_{\alpha,\kappa}(\eta_k, h_k)]\right) \mathrm{KL}\left(\rho_0 \,\|\, \pi\right), \tag{47}$$

$$\mathcal{W}_2^2(\rho_N, \pi) \le \frac{2}{\alpha}\, \mathrm{KL}\left(\rho_N \,\|\, \pi\right). \tag{48}$$

Proof of Theorem B.5. The proof uses exact-Langevin KL decay under LSI (Vempala & Wibisono, 2019, Theorem 4) and Talagrand inequality (Otto & Villani, 2000, Theorem 1), in addition to Theorem B.4.

By Theorem B.4, the one-step bound is

$$\mathrm{KL}\left(\rho_{k+1} \,\|\, \pi\right) \le C_{\alpha,\kappa}(\eta_k, h_k)\, \mathrm{KL}\left(\rho_k \,\|\, \pi\right). \tag{49}$$

We note that $\kappa \ge 0$ and $\kappa h_k(1-\eta_k) \le 2$ imply $\eta_k[1 - \kappa h_k(1-\eta_k)/2] \in [0, 1]$ and $0 < e^{-2\alpha h_k} \le C_{\alpha,\kappa}(\eta_k, h_k) \le 1$. The upper inequality is strict if $\eta_k > 0$ and $\kappa h_k(1-\eta_k) < 2$.

Iterating the one-step descent over k gives the product bound. Applying the inequality $C_{\alpha,\kappa} \le e^{-(1-C_{\alpha,\kappa})}$ to each factor gives the exponential bound in (47).

We now apply Talagrand's inequality

$$\mathcal{W}_2^2(\rho_k, \pi) \le 2\, \mathrm{KL}\left(\rho_k \,\|\, \pi\right)/\alpha. \tag{50}$$

Thus the previous bound provides convergence of both KL and $\mathcal{W}_2$. □

Corollary 4.3 (Geometric convergence under LSI and bounded negative curvature)**.** *Under the notations of §4.1, and the α-LSI and curvature assumptions above, with fixed $h_k = h > 0$ and $\eta_k = \eta \in (0, 1]$ with $\kappa h(1-\eta) < 2$, $\mathcal{W}_2^2(\rho_k, \pi)$ and $\mathrm{KL}\left(\rho_k \,\|\, \pi\right)$ converge geometrically to 0.*

Proof of Corollary 4.3. Immediate from Theorem B.5, as both discrepancies tend to zero if $\sum_{k\ge 0}[1 - C_{\alpha,\kappa}(\eta_k, h_k)] = \infty$. □

C Extended practical considerations

This appendix details the main design choices of OSIS. It expands on the kernel, loss, tempering, and inference choices in §5. This section aims to provide a general overview. Specific details about the experiments are provided in §D.

C.1 Choice of MCMC kernel

We present here various methods to update the generated samples towards the target π. It often consists of a π-invariant MCMC kernel, but other methods such as *Stein variational gradient descent* (Liu & Wang, 2016) can also be investigated. In this subsection, $q(y \mid x)$ denotes a proposal density. We first describe kernels for a fixed target π. Under an annealing schedule, the same constructions use π_{β_t} instead.

Metropolis-Hastings. The Metropolis-Hastings (MH Metropolis et al., 1953; Hastings, 1970) kernel modifies a continuous full-support proposal to be π-invariant by mixing it with a 'rejection' kernel. To be precise, given a proposal density q, define

$$K(y \mid x) = \alpha(x, y)q(y \mid x) + (1 - A(x))\delta_x(y), \tag{51}$$

where δ_x corresponds to a Dirac delta at x and

$$\alpha(x, y) = \min\left(1, \frac{\pi(y)q(x \mid y)}{\pi(x)q(y \mid x)}\right) \text{ and } A(x) = \int \alpha(x, y)q(y \mid x)\,\mathrm{d}y. \tag{52}$$

By construction, K is π-invariant (in particular, if q is reversible with respect to π, then $A(x) = 1$ for π-almost every x and $K = q$). However, the choice of q greatly affects the rate of convergence to π of a Markov chain with transition operator K.

Random-walk proposals. Random-walk Metropolis (RWM) is a commonly used form of MH. One such proposal is a Gaussian random walk that takes $q(y \mid x) = \mathcal{N}(y; x, \sigma^2 I)$ for some chosen variance σ^2. The proposal is symmetric, so its acceptance probability simplifies to $\min(1, \pi(y)/\pi(x))$. RWM requires target-energy evaluations but no target gradients.

Langevin-based proposals. One can make use of target gradient information when defining a proposal distribution. Metropolis-Adjusted Langevin (MALA; Rossky et al., 1978; Besag, 1994; Roberts & Tweedie, 1996) is MH based on a proposal q that is an Euler-Maruyama integration step of a Langevin SDE:

$$q(y \mid x) = \mathcal{N}\left(y; x - \delta\nabla\mathcal{E}(x), 2\delta I_d\right)$$

for some chosen step size $\delta > 0$. At inverse temperature β_t, the drift becomes $-\delta\beta_t\nabla\mathcal{E}(x)$. Unlike RWM, this proposal is generally asymmetric, so both proposal-density terms in (52) must be retained. Omitting the accept/reject step gives the unadjusted Langevin algorithm (ULA), which does not in general preserve the target at a finite step size (Roberts & Tweedie, 1996). Neither a MALA step nor a ULA step is the exact Langevin semigroup transition K^h used in §4.

Higher-order splitting schemes. Splitting integrators provide further discretisations of Langevin dynamics and can reduce finite-step sampling bias (Leimkuhler & Matthews, 2013a). The idea is to divide the dynamics into simpler parts that can each be integrated exactly, then compose their updates. We consider the matrix formulation of the underdamped Langevin targeting $\pi(x)\mathcal{N}(v; 0, I_d)$, where $\pi \propto e^{-\mathcal{E}}$:

$$\begin{bmatrix}\mathrm{d}X_s \\ \mathrm{d}V_s\end{bmatrix} = \underbrace{\begin{bmatrix}V_s \\ 0\end{bmatrix}\mathrm{d}s}_{\text{A}} + \underbrace{\begin{bmatrix}0 \\ -\nabla\mathcal{E}(X_s)\end{bmatrix}\mathrm{d}s}_{\text{B}} + \underbrace{\begin{bmatrix}0 \\ -\zeta V_s\,\mathrm{d}s + \sqrt{2\zeta}\,\mathrm{d}B_s\end{bmatrix}}_{\text{O}}. \tag{53}$$

where $\zeta > 0$ is the friction coefficient and $v \in \mathbb{R}^d$ is the auxiliary velocity variable. The underdamped Langevin dynamics is split into three parts: free transport of the position at fixed velocity (A), force-driven evolution of the velocity at fixed position (B), and Ornstein-Uhlenbeck evolution of the velocity, combining friction and Gaussian noise at fixed position (O). Their exact updates over a

duration $t \geq 0$ are

$$\mathrm{A}_t : \quad (x, v) \longmapsto (x + tv, v), \tag{54}$$

$$\mathrm{B}_t : \quad (x, v) \longmapsto (x, v - t\nabla\mathcal{E}(x)), \tag{55}$$

$$\mathrm{O}_t : \quad (x, v) \longmapsto \left(x, e^{-\zeta t} v + \sqrt{1 - e^{-2\zeta t}}\, \xi\right), \qquad \xi \sim \mathcal{N}(0, I_d). \tag{56}$$

These exact substeps generally do not commute, so their composition provides a specific approximation of the full dynamics (Leimkuhler & Matthews, 2013b, Section IV). For example, the chronological order of a BAOAB step of size δ is

$$\mathrm{B}_{\delta/2} \longrightarrow \mathrm{A}_{\delta/2} \longrightarrow \mathrm{O}_{\delta} \longrightarrow \mathrm{A}_{\delta/2} \longrightarrow \mathrm{B}_{\delta/2}. \tag{57}$$

The letters specify the order of the updates; each repeated letter here uses a half step. Each update acts on the current state, with independent noise for each O substep. Symmetric Strang composition (*i.e.*, letters forming a palindrome) give second-order weak accuracy under appropriate conditions (Strang, 1968).

HMC. Hamiltonian Monte Carlo introduces auxiliary momentum variables and uses Hamiltonian trajectories, with a correction for numerical integration error, to construct a target-invariant transition (Neal, 2011).

Stein variational gradient descent. Stein variational gradient descent (SVGD) transports a set of interacting particles using a deterministic, kernel-based field (Liu & Wang, 2016). For a differentiable positive-definite kernel k and step size $\delta > 0$, its update for particles $\{x_i\}_{i=1}^{B}$ is

$$y_i = x_i + \frac{\delta}{B} \sum_{j=1}^{B} \left[-k(x_j, x_i)\nabla\mathcal{E}(x_j) + \nabla_{x_j} k(x_j, x_i)\right], \qquad i = 1, \ldots, B. \tag{58}$$

Multiple-step kernels. If K is a π-invariant kernel, so is K^n for $n \geq 1$. Thus one can construct several-step kernels by iterating the primitive ones defined above, as we do in the experiments.

Acceptance-targeting step adaptation. We adapt the step size δ of random-walk or Langevin MCMC during training to maintain a target acceptance rate $\alpha^\star$ (Andrieu & Thoms, 2008). The update is log-space stochastic approximation,

$$\log\delta \leftarrow \log\delta + \eta_{\text{adapt}}(\hat{\alpha} - \alpha^\star),$$

where $\hat{\alpha}$ is the batch mean acceptance. To avoid relearning the proposal scale whenever β_t changes, the default adaptive state is $h_t := \frac{\delta}{\beta_t}$. The per-iteration value $\delta_t = h_t\beta_t$ is then clipped to $[\delta_{\min}, \delta_{\max}]$. This keeps the effective MALA scale invariant across temperature transitions.

C.2 Choice of divergence

To compute the distance between two distributions, we rely on sample-based distribution metrics. In practice, these distances are computed using samples from both distributions.

Maximum mean discrepancy. One possible way to compute the distance between two distributions is use maximum mean discrepancy (MMD; Gretton et al., 2012). Given a kernel function, such as $k(x, y) = \exp\left(-\frac{\|x-y\|}{h}\right)$, where h is a positive bandwidth and $\|\cdot\|$ is L_2 distance, the MMD between two distributions μ_1 and μ_2 is defined as:

$$\begin{aligned}\mathrm{MMD}^2(\mu_1, \mu_2) := \mathbb{E}_{x,x'\sim\mu_1}[k(x, x')] + \mathbb{E}_{y,y'\sim\mu_2}[k(y, y')] \\ - 2\mathbb{E}_{x\sim\mu_1, y\sim\mu_2}[k(x, y)],\end{aligned} \tag{59}$$

which can be estimated by Monte Carlo sampling.

Squared 2-Wasserstein distance. For probability distributions μ_1 and μ_2 on $\mathbb{R}^d$ with finite second moments, the squared 2-Wasserstein distance is an optimal transport cost

$$\mathcal{W}_2^2(\mu_1, \mu_2) := \left\{\inf \int_{\mathbb{R}^d\times\mathbb{R}^d} |x - y|^2 \gamma(x, y)\, \mathrm{d}x\mathrm{d}y : \gamma \in \Pi(\mu_1, \mu_2)\right\}, \tag{60}$$

where $\Pi(\mu_1, \mu_2)$ contains joint distributions over $\mathbb{R}^d \times \mathbb{R}^d$ whose marginals are μ_1 and μ_2.The $\mathcal{W}_2$ distance between empirical approximations to μ_1 and μ_2 is known as the minibatch 2-Wasserstein, and it converges to (60) as the number of samples increases (Fatras et al., 2021).

Sliced Wasserstein distance. The sliced Wasserstein distance (SWD; Bonneel et al., 2015; Kolouri et al., 2019; Nadjahi et al., 2020) is a tractable surrogate that computes OT distances using 1-dimensional projections. It is defined by

$$\mathbb{S}_2^2(\mu_1, \mu_2) \coloneqq \mathbb{E}_\omega \left[\mathcal{W}_2^2(P_{\omega\#}\mu_1, P_{\omega\#}\mu_2) \right], \tag{61}$$

where the expectation is taken over ω uniform on the unit sphere in $\mathbb{R}^d$. In practice, the expectation over the projection direction ω is estimated with random projections. The quantity inside the expectation is a 1D OT distance between empirical distributions and is thus easily computed.

Sinkhorn divergence. Entropic regularisation is introduced to the transport problem Equation (60) and provides a smoother optimisation problem that can be solved using Sinkhorn iterations. For regularisation strength $\varepsilon > 0$, define (Cuturi, 2013)

$$\mathrm{OT}_\varepsilon(\mu_1, \mu_2) = \inf_{\gamma \in \Pi(\mu_1, \mu_2)} \left\{ \int |x - y|^2 \, \gamma(\mathrm{d}x, \mathrm{d}y) + \varepsilon \, \mathrm{KL}\left(\gamma \,\|\, \mu_1 \otimes \mu_2\right) \right\}. \tag{62}$$

The corresponding debiased Sinkhorn divergence subtracts the two self-interaction terms,

$$\mathrm{S}_\varepsilon(\mu_1, \mu_2) = \mathrm{OT}_\varepsilon(\mu_1, \mu_2) - \tfrac{1}{2}\,\mathrm{OT}_\varepsilon(\mu_1, \mu_1) - \tfrac{1}{2}\,\mathrm{OT}_\varepsilon(\mu_2, \mu_2), \tag{63}$$

so that $\mathrm{S}_\varepsilon(\mu, \mu) = 0$. For empirical measures and the squared-Euclidean cost used here, $\mathrm{S}_\varepsilon(\mu_1, \mu_2)$ approaches $\mathcal{W}_2^2(\mu_1, \mu_2)$ as $\varepsilon \downarrow 0$. As $\varepsilon \to \infty$, it approaches the squared distance between the means of μ_1 and μ_2, equivalently the squared MMD associated with the linear kernel $k(x, y) = \langle x, y \rangle$ (Feydy et al., 2019, Section 1.2).

D Detailed experiment setup

D.1 Synthetic targets

Setup. We train OSIS and the baselines for five random seeds. After training, we draw five independent batches of 10,000 samples with different seeds at the selected checkpoint: the final training iteration for OSIS and the checkpoint with the highest ELBO for baselines, following Blessing et al. (2024).

Each sampler's evaluation batch is compared with 10,000 reference samples fixed for all samples. We average the results within each training seed, then report the mean and sample standard deviation across the five training-seed means. SMC uses five repetition groups instead of training seeds. The reference rows provide a floor on each metric by comparing two independent target batches of samples. We report empirical $\mathcal{W}_2^2$ with unnormalised squared Euclidean cost, computed using POT (Flamary et al., 2021) with convergence checks. MMD corresponds to the square root of the biased Gaussian kernel MMD-squared estimate, with bandwidth fixed per target to the median pairwise distance in an independent 10,000-sample batch from the target distribution.

Baselines. We retrain FAB (Midgley et al., 2023), PIS (Zhang & Chen, 2022), and DIS (Berner et al., 2024), and run SMC (Del Moral et al., 2006) using the reference implementation of Blessing et al. (2024). FAB uses an 8-layer Real-NVP flow with $[64, 64]$ conditioner MLPs, 12 intermediate distributions, 10 HMC transitions per intermediate distribution, a replay buffer, and α-divergence loss with $\alpha = 2$. It trains for 3,000 iterations with four replay updates per iteration, at batch size 2,000 (1,000 for both NICE targets).

PIS and DIS use the PISGRADNet with width-64 MLPs and Fourier time features, trained for 40,000 updates with a batch size of 2,000 and 128 integration steps. For other hyperparameters, we use those reported in Blessing et al. (2024). For Manywell $d = 32, 64$ and 40GMM $d = 10$, since the paper does not provide default hyperparameters, we performed a grid search over the key hyperparameters, as provided in Table 5, and selected the one with the best ELBO. SMC uses 2,000 particles, 10 HMC transitions per annealing interval, and resampling threshold of 0.3. It uses 256 temperatures for GMM-10D and ManyWell (dimension 32 and 64) targets, and 128 otherwise.

Table 5: Baseline hyperparameter search grids.

Method	Parameter	GMM ($d = 10$)	ManyWell ($d = 32, 64$)
FAB / PIS / DIS	Learning rate	$\{10^{-5}, 10^{-4}, 5 \cdot 10^{-4}, 10^{-3}\}$	
FAB	Base scale	$\{20, 40, 80\}$	$\{1, 2, 4\}$
DIS	Initial std.	$\{20, 40, 80\}$	$\{1, 2, 4\}$
PIS	Max. diffusion amplitude	$\{80, 160, 320\}$	$\{2, 4, 8\}$
SMC	Initial std.	$\{20, 40, 80\}$	$\{1, 2, 4\}$
	HMC step size	$\{0.1, 0.2\}$	$\{0.05, 0.1\}$
	Number of temperatures	$\{128, 256\}$	$\{128, 256\}$

Table 6: OSIS hyperparameters. ResNN(hidden, blocks) and UNet(base channels, attention heads). The temperature schedule is as defined in §D.

Target	Loss	Model	lr	τ_{start}	target accept.
40GMM ($d = 2$)	SWD	ResNN(256, 3)	$3 \cdot 10^{-3}$	128	0.3
40GMM ($d = 10$)	SWD	ResNN(256, 3)	$3 \cdot 10^{-3}$	256	0.3
ManyWell ($d = 32$)	SWD	ResNN(256, 3)	$3 \cdot 10^{-3}$	256	0.90
ManyWell ($d = 64$)	SWD	ResNN(256, 3)	$3 \cdot 10^{-3}$	16	0.574
Funnel ($d = 10$)	SWD	ResNN(128, 2)	$3 \cdot 10^{-5}$	1	0.90
NICE-MNIST ($d = 196$)	SWD	UNet(16, 4)	$1 \cdot 10^{-3}$	32	0.574
NICE-FMNIST ($d = 784$)	SWD	UNet(16, 4)	$1 \cdot 0^{-4}$	32	0.574

For TB+MALA (Sendera et al., 2024) and SubTB+SMC (Choi et al., 2026), we use the original implementation of Choi et al. (2026) and reimplement TB+MALA within the same codebase to simplify benchmarking. While the reimplemented TB+MALA is not identical to the original from Sendera et al. (2024), we ensure that the algorithm performs at least as well as, or better than, the original on synthetic targets. For these two baselines, we set the network architectures and common hyperparameters (learning rate and initial std.) to match those of DIS, while fixing algorithm-specific hyperparameters as reported in the original papers.

OSIS. For the main evaluation study Tables 1 and 2, OSIS was trained using a squared SWD (Bonneel et al., 2015) with 2048 projections, AdamW with weight decay 10^{-2}, batch size of 4096, $T = 10{,}000$ training updates, 20 steps of MCMC refinement and $M = 50$ inner gradient iterations Algorithm 3. All the other experiments were run for $M = 1$.

The OSIS architecture uses a ResNN (He et al., 2016) on the synthetic targets and a small UNet on the NICE targets (Ronneberger et al., 2015). The temperature follows the schedule of Fogliani et al. (2026): at time t, the inverse temperature is

$$\beta(t) = \beta_0^{\max(0,\, 1 - t/(0.8T))},$$

so the last 20% of training uses $\beta = 1$. Funnel keeps $\beta = 1$ during its entire training. Additional hyperparameters are reported in Table 6.

Kernel comparison. For this study the model is kept fixed across kernels as described above. Each kernel uses 20 refinement steps. Other parameters are tuned per kernel and presented in Table 7. MALA in this study uses $M = 1$ inner iterations (see Algorithm 3), so its parameters differ from Table 6.

Parameter counts. Trainable parameter counts are reported in Table 8. SMC is non-amortised and has no learned parameters. PIS, DIS, and TB+MALA share the same backbone and parameter count. OSIS uses a ResNN on the synthetic targets and a UNet on the NICE targets.

Sampling time. Figure 3 provides the median time over 250 runs for a batch of 2,000 samples on one NVIDIA GH200 GPU, excluding compilation and warm-up. The mean $\mathcal{W}_2^2$ metric is computed independently following the previously described 10,000-sample protocol. OSIS, TB+MALA, and SubTB+SMC use compiled PyTorch, while FAB, PIS, DIS and SMC use native JAX JIT.

Table 7: Selected hyperparameters for ManyWell ($d = 32$).

Kernel	Learning rate	Initial temperature	Initial proposal standard deviation	Target acceptance rate	ULA drift step	Integration step size	Friction	Leapfrog steps	SVGD update step	RBF kernel variance
RWMH	10^{-3}	4	1	0.5	—	—	—	—	—	—
MALA	10^{-3}	16	—	0.90	—	—	—	—	—	—
ULA	10^{-3}	16	—	—	0.008	—	—	—	—	—
BAOAB	$1.9 \cdot 10^{-5}$	2	—	—	—	0.0025	2	—	—	—
OBABO	10^{-3}	16	—	—	—	0.006	2	—	—	—
HMC	$1.7 \cdot 10^{-3}$	16	—	—	—	0.019	—	10	—	—
SVGD	10^{-3}	16	—	—	—	—	—	—	$1.6 \cdot 10^{-4}$	2

Table 8: Number of trainable parameters in the benchmark study ($\times 10^3$).

Target →	40GMM		ManyWell		Funnel	NICE	
Method ↓	$d = 2$	$d = 10$	$d = 32$	$d = 64$	$d = 10$	MNIST	FMNIST
SMC	—	—	—	—	—	—	—
FAB	70.7	83.1	117.2	166.9	83.1	371.8	1284.4
PIS / DIS	37.8	39.3	43.6	49.8	39.3	75.4	189.5
OSIS	198.7	202.8	214.0	230.5	35.7	227.9	689.9

D.2 Outsourced sampling

Target densities. For MNIST we use a VAE (Kingma & Welling, 2014) which we train ourselves. Then, we use VAE decoder with a 16-dimensional latent, and a Bernoulli likelihood, trained for 100 epochs. For the reward function, we train a CNN classifier $p(c \mid x)$. The reward is defined as $\log r(z) = \log \sum_c \tilde{w}_c, p(c \mid x(z))$, and the target log-density is $\log \pi(z) = \log r(z) - \frac{1}{2}|z|^2$. Both networks remain frozen during the training of OSIS.

For the experiments on FFHQ-64, we use a pretrained frozen R3GAN (Huang et al., 2024) generator. The reward is ImageReward (Xu et al., 2023). The target log-density is defined as $\log \pi(z) = 3.0 \cdot \mathrm{IR}(G(z), \text{caption}) - \frac{1}{2}|z|^2$. Reward and gradient passes run `bfloat16`.

Model architecture for G_θ. In all our experiments we use a ResNet architecture with MLP layers. The inputs are first projected to a 256-dimensional latent, then follow 3 residual blocks of the form $x \mapsto x + \mathrm{SiLU}(Wx + b)$, $W \in \mathbb{R}^{256\times256}$, then the outputs are projected to the output dimension.

Training. We train OSIS with 1200 outer steps and 50 inner steps, following the procedure presented in Algorithm 3. We use an empirical approximation of OT loss (§C.2) and use MALA with adaptive step size (§C.1) as the MCMC algorithm, which we run for 20 steps. For the initial 1000 outer iterations we train the algorithm on a tempered target, the temperature follows a cosine schedule with initial temperature 2.0 and final temperature 1.0. Next, we train OSIS for 200 more iterations keeping the temperature fixed at 1.0. We use the AdamW (Loshchilov & Hutter, 2019) optimiser with learning rate 0.001 and no weight decay.

Evaluations. Each method in Table 4 is evaluated using $N = 10{,}000$. The reference is rejection sampling from the same tilted density. Since ImagerReward is unbounded by default, we bound it with a constant to perform rejection sampling. $\mathbb{E}[\log r(x)]$ corresponds to the averaged ImageReward score computed with $10,000$ samples. FID is computed against the rejection-sampling reference. We compute FID score using `torchmetrics`.

E Additional experiment results

E.1 Extended evaluation of OSIS

In this section, we report results for OSIS with OBABO kernels on the main synthetic targets. This provides further evidence on whether splitting schemes can improve OSIS performance Tables 9 and 10. Overall, the benefits depend on the training objective: with OT, OBABO achieves lower mean squared 2-Wasserstein distance and MMD than MALA on all evaluated targets, whereas with SWD, the results are mixed. These experiments compare equal numbers of OSIS training steps, which does

Table 9: Sampler comparison on GMM, Funnel, and ManyWell targets. OSIS rows specify the training kernel and loss. This table was computed using $M = 1$ inner iteration (see Algorithm 3).

Target →	40GMM ($d = 2$)		40GMM ($d = 10$)		Funnel ($d = 10$)		ManyWell ($d = 32$)		ManyWell ($d = 64$)	
Method ↓ Metric →	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)
Reference	7.68±3.22	7.17±1.51	202.81±16.77	7.98±0.78	87.60±1.20	9.22±1.54	23.61±0.07	8.75±0.72	78.36±0.11	9.58±0.65
OSIS (OBABO, OT)	81.85±31.72	58.06±15.95	2359.76±535.07	123.10±39.78	189.86±0.33	246.80±0.72	35.54±0.01	255.33±0.34	78.70±0.04	274.45±0.43
OSIS (OBABO, SWD)	848.79±388.69	378.69±132.12	7465.66±7.91	477.83±0.51	99.37±2.09	57.69±8.07	28.77±0.53	85.55±8.86	91.24±2.01	170.13±9.52
OSIS (MALA, OT)	111.24±94.65	83.77±39.73	2913.22±489.84	140.34±31.18	243.15±3.27	474.04±16.40	36.60±0.02	276.63±0.29	81.37±0.10	293.66±0.61
OSIS (MALA, SWD)	553.89±305.14	277.34±116.75	9854.75±1328.11	482.27±132.16	89.41±0.53	45.32±3.21	25.51±0.43	48.57±7.55	89.93±14.74	159.27±20.76
FAB (Midgley et al., 2023)	72.03±19.29	46.42±6.32	6257.04±455.10	380.98±45.19	582.97±233.17	11.87±1.74	25.52±0.30	36.63±5.56	132.55±0.48	299.11±1.51
FAB (flow only) (Midgley et al., 2023)	296.01±333.93	81.05±38.73	6217.94±472.86	382.64±44.65	583.48±233.71	12.94±2.34	27.10±0.46	47.95±4.56	125.58±0.04	401.28±0.64
SMC (Del Moral et al., 2006)	14.67±1.79	16.85±2.11	4362.81±139.06	208.62±11.73	149.56±1.37	125.72±8.99	26.04±0.16	22.48±1.10	85.76±0.29	42.63±1.36
DIS (Berner et al., 2024)	29.26±7.00	34.71±10.99	1650.04±60.09	85.37±5.26	122.58±21.01	214.06±103.49	36.44±2.44	162.78±20.94	100.34±3.52	162.04±18.66
PIS (Zhang & Chen, 2022)	40.35±25.29	34.40±12.85	2702.24±66.69	142.54±5.24	165.77±1.72	194.21±3.72	35.47±0.10	227.64±1.74	83.39±0.13	230.37±1.05

Table 10: Sampler comparison on high-dimensional targets. OSIS rows specify the training kernel and loss. This table was computed using $M = 1$ inner iteration (see Algorithm 3).

Target →	NICE-MNIST ($d = 196$)		NICE-FMNIST ($d = 784$)	
Method ↓ Metric →	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)
Reference	129.56±0.22	8.75±0.08	549.42±1.35	9.24±1.09
OSIS (OBABO, OT)	270.02±4.60	428.07±10.29	1631.93±63.53	412.35±14.51
OSIS (OBABO, SWD)	183.18±5.45	122.34±27.80	2269.09±475.05	410.57±89.89
OSIS (MALA, OT)	277.40±8.45	462.69±18.73	1776.79±82.07	450.47±21.16
OSIS (MALA, SWD)	209.74±39.21	248.96±125.42	2506.97±29.95	507.93±31.28
FAB (Midgley et al., 2023)	143.83±3.10	80.83±14.39	1746.47±335.35	438.79±71.94
FAB (flow only) (Midgley et al., 2023)	162.20±1.91	57.94±11.09	1768.75±335.78	430.25±72.05
SMC (Del Moral et al., 2006)	268.32±4.16	246.17±13.54	2856.36±15.26	629.18±3.24
DIS (Berner et al., 2024)	307.73±55.89	208.21±22.97	5321.68±2656.55	421.53±124.32
PIS (Zhang & Chen, 2022)	237.63±3.08	225.89±16.69	2793.73±19.11	290.08±4.09

not necessarily imply equal computational cost. Each OBABO step requires two energy-gradient evaluations, whereas each MALA step requires two energy-gradient evaluations and two additional energy-value evaluations for the Metropolis–Hastings correction.

E.2 OSIS AGAINST ITS LONG RUN REFINEMENT

In this section we compare OSIS to its long run MCMC kernel. Both methods are run with the same MALA settings, number of MCMC steps and temperature annealing schedule. This table assesses how much of the refinement signal OSIS captures relative to running the equally long rollout of its refinement kernel. We note that the MALA parameters used here come from OSIS runs meaning, that they were not tuned for OSIS training and may not be optimal for direct MCMC sampling.

We observe that OSIS has higher reported discrepancies than MALA on 40GMM, motivating further investigation of whether additional inner optimisation steps help OSIS track its refinement targets Table 1. On ManyWell, however, SWD-trained OSIS outperforms its refinement kernel run for the same number of steps. These results show effective amortisation of the refinement signal in our sampler on specific targets.

Table 11: Comparison of OSIS +MALA with long-run MALA only. OSIS rows specify the training kernel and loss. MALA-only runs start from Gaussian noise. We compare one set of MALA samples against five reference sets and report the average and standard deviation. This table was computed using $M = 1$ inner iteration (see Algorithm 3).

Target →	40GMM ($d = 2$)		40GMM ($d = 10$)		Funnel ($d = 10$)		ManyWell ($d = 32$)		ManyWell ($d = 64$)	
Method ↓ Metric →	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)
Reference	7.68±3.22	7.17±1.51	202.81±16.77	7.98±0.78	87.60±1.20	9.22±1.54	23.61±0.07	8.75±0.72	78.36±0.11	9.58±0.65
OSIS (MALA, OT)	111.24±94.65	83.77±39.73	2913.22±489.84	140.34±31.18	243.15±3.27	474.04±16.40	36.60±0.02	276.63±0.29	81.37±0.10	293.66±0.61
OSIS (MALA, SWD)	553.89±305.14	277.34±116.75	9854.75±1328.11	482.27±132.16	89.41±0.53	45.32±3.21	25.51±0.43	48.57±7.55	89.93±14.74	159.27±20.76
MALA only	82.82±2.28	52.04±1.67	200.14±12.26	8.58±0.62	90.71±3.41	22.47±3.52	36.58±0.05	149.82±0.74	104.71±0.11	166.18±0.86

E.3 DRIFT PARAMETERISATION OF OSIS

We evaluate whether OSIS can benefit from a drift-based regression formulation motivated by Wasserstein gradient flows. This formulation expresses the generator update as a regression task towards samples displaced by an estimated drift field, with the regression targets held fixed during differentiation. For OT-based objectives, computing these targets still requires the corresponding transport information. The drift parameterisation yields lower mean values for both metrics on ManyWell ($d = 32$). On 40GMM ($d = 2$), both mean Sinkhorn divergence and mean MMD are higher, although the differences are within the scale of the reported variability.

Table 12: Comparison of discrepancy and drift objectives for OSIS trained with the OBABO refinement kernel. Results are mean ± standard deviation over five training seeds, after averaging five sampling-seed evaluations per checkpoint. This table was computed using $M = 1$ inner iteration (see Algorithm 3).

Target →	40GMM ($d = 2$) + OT		ManyWell ($d = 32$) + SWD	
Method ↓ Metric →	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)	$\mathcal{W}_2^2$	MMD ($\times 10^{-3}$)
Reference	$7.68_{\pm 3.22}$	$7.17_{\pm 1.51}$	$23.61_{\pm 0.07}$	$8.75_{\pm 0.72}$
Discrepancy	$81.85_{\pm 31.72}$	$58.06_{\pm 15.95}$	$28.77_{\pm 0.53}$	$85.55_{\pm 8.86}$
Drift	$95.93_{\pm 46.34}$	$77.97_{\pm 35.87}$	$27.11_{\pm 1.29}$	$69.34_{\pm 13.05}$

E.4 Ablations

We provide in this section detailed ablations of the different design choices of OSIS.

Table 13: Ablation on ManyWell ($d = 32$). The default OSIS training configuration uses 20-step MALA, batch size 4096, SWD-2048, and 50,000 training updates. Results are mean ± standard deviation over five training seeds, after averaging five sampling-seed evaluations per checkpoint. This table was computed using $M = 1$ inner iteration (see Algorithm 3).

Configuration	$\mathcal{W}_2^2$ (↓)	MMD ($\times 10^{-3}$)
Default	$25.51_{\pm 0.43}$	$48.57_{\pm 7.55}$
MCMC steps (default: 20)		
1	$28.40_{\pm 1.19}$	$77.78_{\pm 20.34}$
5	$30.17_{\pm 1.32}$	$109.10_{\pm 29.69}$
10	$27.98_{\pm 1.87}$	$85.43_{\pm 41.68}$
40	$25.30_{\pm 0.55}$	$51.75_{\pm 12.59}$
80	$25.91_{\pm 0.33}$	$43.79_{\pm 9.80}$
Batch size (default: 4096)		
1024	$26.37_{\pm 1.19}$	$60.70_{\pm 16.79}$
2048	$25.58_{\pm 0.32}$	$47.69_{\pm 5.16}$
8192	$24.96_{\pm 0.59}$	$48.07_{\pm 16.14}$
SWD projections (default: 2048)		
16	$29.24_{\pm 0.34}$	$150.38_{\pm 6.05}$
64	$25.63_{\pm 0.20}$	$82.13_{\pm 3.71}$
256	$24.88_{\pm 0.32}$	$52.94_{\pm 12.83}$
1024	$25.01_{\pm 0.32}$	$48.03_{\pm 4.93}$
4096	$25.36_{\pm 0.43}$	$45.81_{\pm 8.22}$
Divergence (default: SWD-2048)		
Exact OT	$36.60_{\pm 0.02}$	$276.63_{\pm 0.29}$
MMD	$36.70_{\pm 0.05}$	$230.28_{\pm 1.27}$

E.5 Additional samples from Bayesian posteriors on MNIST

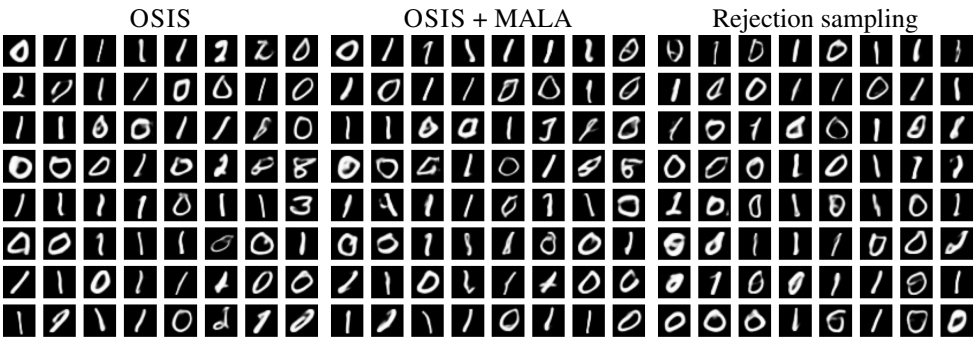

Figure 6: Samples obtained using outsourced sampling. Samples correspond to the classifier-based target with the reward $r(x) = 0.5p(0|x) + 0.5p(1|x)$. From left to right, samples are obtained with OSIS, OSIS with 20-step MALA refinement, and rejection sampling.

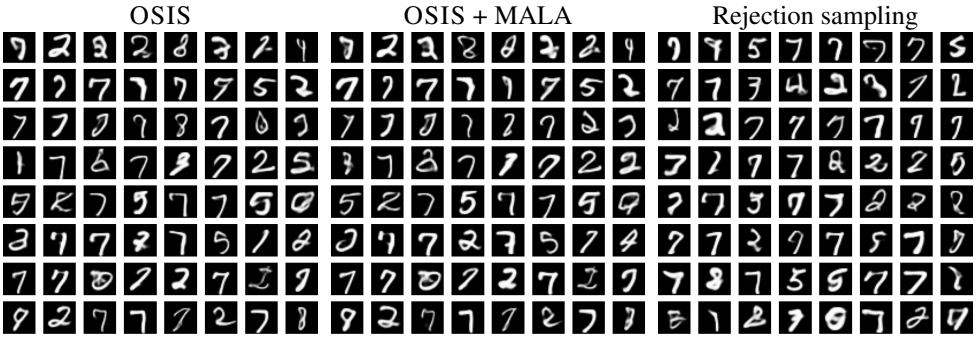

Figure 7: Samples obtained using outsourced sampling. Samples correspond to the classifier-based target with the reward $r(x) = 1p(2 \mid x) + 0.5p(5 \mid x) + 2p(7 \mid x)$. From left to right, samples are obtained with OSIS, OSIS with 20-step MALA refinement, and rejection sampling.

E.6 Additional samples from Bayesian posteriors on FFHQ-64

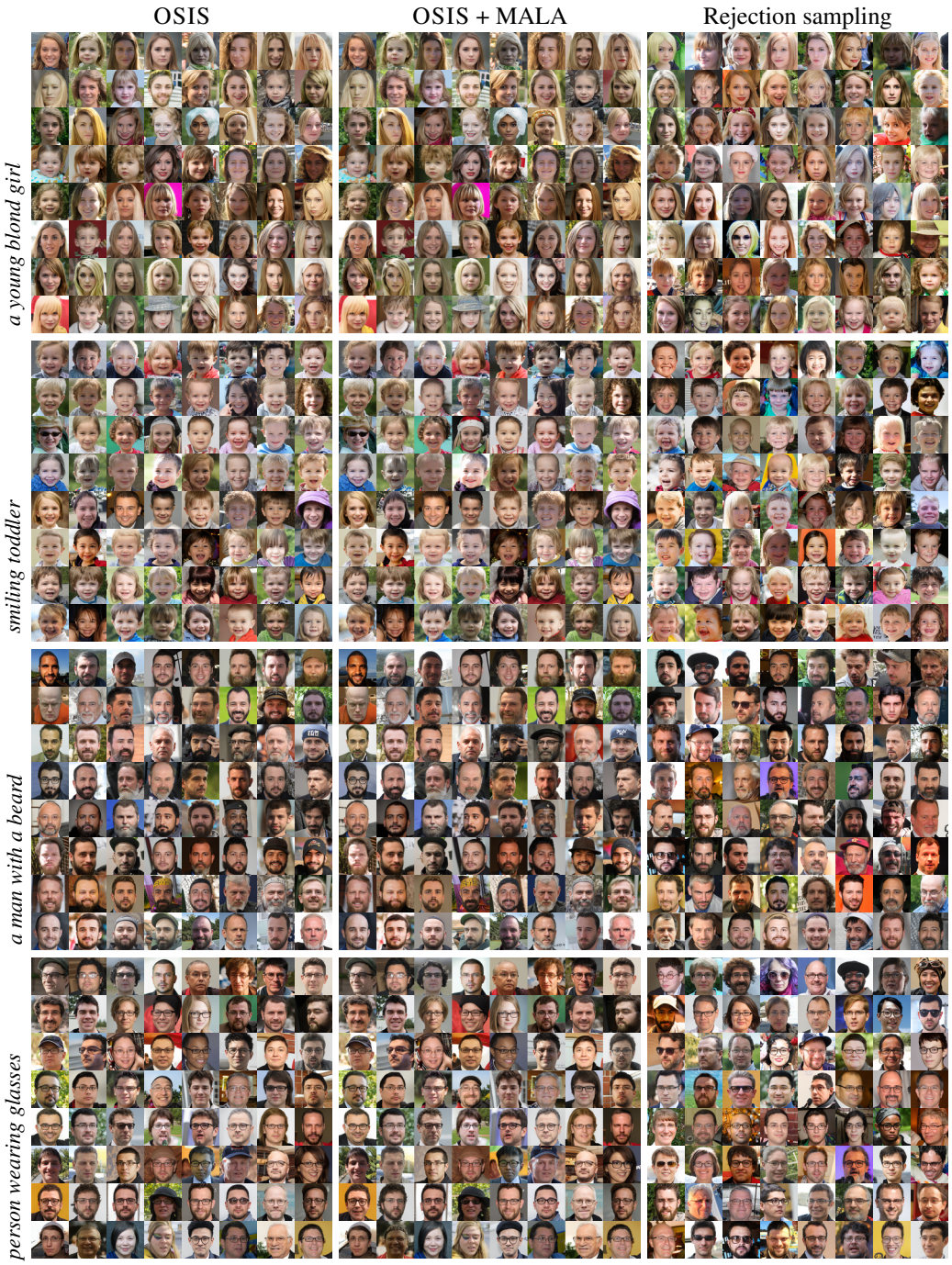

Figure 8: Comparison of samples obtained with OSIS, OSIS with 20-step MALA refinement, and rejection sampling on outsourced sampling task with ImageReward-based target. The following 4 prompts are considered (from to to bottom): *a young blond girl*, *smiling toddler*, *a man with a beard*, *person wearing glasses*.



Figure 9: Visualisation of samples from OSIS, rejection sampling, TB+MALA (Sendera et al., 2024) and MALA. All algorithms, except rejection sampling, are matched in terms of the number of log-density evaluations.