# To discretize continually: Mean shift interacting particle systems for Bayesian inference

**Ayoub Belhadji*** **Daniel Sharp*** **Youssef Marzouk**

Center for Computational Science and Engineering
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology

## Abstract

Integration against a probability distribution given its unnormalized density is a central task in Bayesian inference and other fields. We introduce new methods for approximating such expectations with a small set of weighted samples—i.e., a quadrature rule—constructed via an interacting particle system that minimizes maximum mean discrepancy (MMD) to the target distribution. These methods extend the classical mean shift algorithm, as well as recent algorithms for optimal quantization of empirical distributions, to the case of continuous distributions. Crucially, our approach creates dynamics for MMD minimization that are invariant to the unknown normalizing constant; they also admit both gradient-free and gradient-informed implementations. The resulting mean shift interacting particle systems converge quickly, capture anisotropy and multi-modality, avoid mode collapse, and scale to high dimensions. We demonstrate their performance on a wide range of benchmark sampling problems, including multi-modal mixtures, Bayesian hierarchical models, PDE-constrained inverse problems, and beyond.

## 1 Introduction

The need to compute expectations with respect to a target probability distribution $\pi$ is ubiquitous in statistical inference and machine learning. Most distributions, however, do not admit any closed-form numerical integration technique. This situation often arises in Bayesian inference, where posterior quantities of interest are expressed as integrals over a distribution whose density is only known up to normalization; similar situations arise in computational chemistry and other physical sciences.

Recent work has approached integral discretization through gradient flows on the space of probability measures [71, 21]. In this framework, a discretization is identified with a probability measure, often an empirical measure supported on finitely many particles, and is characterized as the solution of a variational problem. One then constructs a gradient flow to minimize the chosen discrepancy between this measure and the target distribution $\pi$. Such formulations naturally give rise to partial differential equations (PDEs) which may be discretized via systems of interacting particles [59, 32, 33, 16, 63, 76].

Crucially, the most useful gradient flow methods have dynamics invariant to the unknown normalizing constant of the target density; this feature is usually due to the choice of discrepancy, e.g., Kullback–Leibler (KL) divergence, $\chi^2$ divergence, or kernelized Stein discrepancy (KSD). Designing particle systems as discretizations of gradient flows leads to algorithms with strong theoretical guarantees largely concerned with the mean-field limit, i.e., when the number of particles goes to infinity. The

*Equal contributions. Author emails: abelhadj@mit.edu, dannys4@mit.edu, ymarz@mit.edu

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

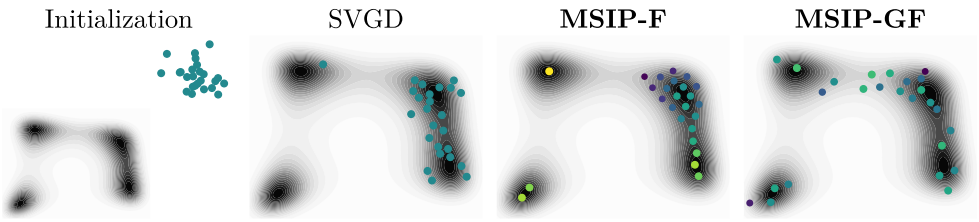


Figure 1: Example of our methods, with color indicating particle weight. (Left): Particle initialization far from support. (Middle left): Result of SVGD with 100 target gradient evaluations per particle. (Middle right): Our gradient-informed method with 100 target gradient evaluations per particle. (Right): Our gradient-free method with 500 target density evaluations per particle.

practical, non-asymptotic regime (with finite particles) still presents numerous challenges, especially for target distributions exhibiting strong non-Gaussianity, local anisotropy, and multi-modality.

In this paper, we study integration through the lens of *quadrature rules*. A quadrature rule approximates the expectation of a function $f : \mathcal{X} \to \mathbb{R}$ by a weighted average of function evaluations,

$$\mathbb{E}_\pi[f(Y)] = \int_{\mathcal{X}} f(y)\,\mathrm{d}\pi(y) \approx \sum_{i=1}^{M} w_i f(y_i), \tag{1}$$

where the weights $w_i \in \mathbb{R}$ and the nodes $y_i \in \mathcal{X}$. From this perspective, most interacting particle systems use *uniform* weights, a design choice that constrains their effectiveness as numerical integration schemes. We instead develop *weighted* quadrature rules for essentially arbitrary unnormalized densities. This task is nontrivial, as classical quadrature constructions rely on estimating high-order moments [39, 34], which is generally intractable for unnormalized densities.

We show that effective quadrature rules for unnormalized densities can be designed by minimizing a regularized maximum mean discrepancy (MMD) associated with the squared exponential kernel. This result is unexpected: MMD minimization is typically viewed as intractable for high-dimensional unnormalized densities. Our approach circumvents this obstacle by introducing a damped fixed-point iteration whose dynamics are invariant, by construction, to the unknown normalizing constant. Our main contributions are as follows:

- We extend an optimal quantization method from the setting of empirical target distributions to continuous target distributions known through their unnormalized densities, as in sampling and Bayesian inference.
- We introduce finite-particle dynamics for MMD minimization in this setting, and prove that they are invariant to the normalization of the target despite the fact that MMD itself depends on this constant.
- We show that certain regularization techniques needed for stability of our kernel implementation are cleanly interpretable as optimization of a *regularized* MMD.
- We introduce multiple *gradient-free* and *gradient (score)-informed* formulations of our algorithm, suitable for a broad range of application domains.
- We demonstrate that our approach is scalable to high dimensions and to diverse, geometrically challenging (e.g., strongly multi-modal and anisotropic) target distributions.

**Notation** For a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ with elements $[\boldsymbol{A}]_{ij} = a_{ij}$, we write the matrix slice as $[\boldsymbol{A}]_{i,:} := (a_{i1}, a_{i2}, \ldots, a_{in})$. Throughout the paper, we consider a domain of interest $\mathcal{X} \subseteq \mathbb{R}^d$ and a density $\pi : \mathcal{X} \to \mathbb{R}^+$. Then, we use a set of particles $\{y_1, \ldots, y_m\} \subset \mathcal{X}$ in a configuration denoted $Y \in \mathcal{X}^M$ with $[Y]_{i,:} := y_i$. For $\mathbf{x} \in \mathbb{R}^M$, we use the notation $\mathrm{diag}(\mathbf{x}) \in \mathbb{R}^{M\times M}$ to denote the diagonal matrix whose $(i,i)$-th entry is $x_i$. Given a symmetric positive-definite kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$, we define the kernel matrix $\boldsymbol{K} : \mathcal{X}^M \to \mathbb{R}^{M\times M}$ via $[\boldsymbol{K}(Y)]_{ij} = \kappa(y_i, y_j)$. We write the *kernel mean embedding* $\boldsymbol{v}_0 : \mathcal{X}^M \to \mathbb{R}^M$ of a density $\pi$ as

$$v_0(y) = \int \kappa(x, y)\ \mathrm{d}\pi(x), \quad [\boldsymbol{v}_0(Y)]_i = v_0(y_i). \tag{2}$$

Similarly, we define the *kernel embedding of the first moment* $\boldsymbol{v}_1 : \mathcal{X}^M \to \mathbb{R}^{M\times d}$ as

$$v_1(y) = \int x\kappa(x,y)\ \mathrm{d}\pi(x), \quad [\boldsymbol{v}_1(Y)]_{i,:} = v_1(y_i). \tag{3}$$

## 2 Theory and background

We contextualize our approach by reviewing literature on integration given an unnormalized density.

### 2.1 From MCMC methods and Langevin dynamics to interacting particle systems

The overdamped Langevin diffusion admits an invariant law with density $\pi$. Discretizing this diffusion leads to Markov chain Monte Carlo (MCMC) algorithms such as ULA and MALA [12, 69], which incorporate gradients of the log-density and enable more efficient exploration. In the latter case, Metropolization [65, 68] ensures that the target distribution is preserved after discretization. There are of course myriad other sampling algorithms using Markov chains for Bayesian inference and generative modeling, including both gradient-free [40, 42] and gradient-based [28, 48] approaches. Yet these algorithms often struggle on the geometrically complex target distributions described in §1.

Jordan et al. [51] interpret the Langevin dynamics as the gradient flow of the KL divergence in the Wasserstein geometry. The gradient flow is expressed as a PDE governing the evolution of a probability density $\rho_t$ for which $\mathrm{KL}(\rho_t\|\pi)$ decreases with $t$. This variational point of view has since been extended to alternative geometries and divergences. These variational approaches emphasize using an *ensemble of interacting particles* instead of independent Markov chains.

**Stein variational gradient descent** SVGD is derived as a gradient flow of the KL divergence under the Stein geometry [29], which is induced by a reproducing kernel Hilbert space (RKHS) [59]. At each iteration, a nonlinear map defined by the kernel of the RKHS is applied to the set of particles. This algorithm has since been the subject of extensive theoretical investigation. Convergence of mean-field, i.e., infinite-particle, SVGD dynamics in various divergences and discrepancies has been discussed thoroughly [58, 55, 20, 70, 74, 29, 45]. More recently, [72, 61, 25, 6] obtain convergence guarantees for the finite-particle SVGD algorithm. Despite this theory, standard SVGD empirically suffers from mode collapse, often failing to cover all modes of multi-modal target distributions [78].

**Interacting particle Langevin samplers** Inspired by methods for Bayesian filtering [30], ensemble Kalman approaches have been introduced for gradient-free sampling in the Bayesian setting, particularly with Gaussian observation models. In the mean-field limit, certain ensemble Kalman samplers (EKS) realize gradient flows of the KL divergence with respect to the so-called Kalman–Wasserstein geometry. The resulting PDE is a Fokker–Planck equation preconditioned by the covariance matrix of the evolving density [32]. EKS also admits affine-invariant variants, e.g., ALDI [33], a crucial property for highly anisotropic target densities where standard Langevin dynamics converge slowly.

**Kernel Fisher–Rao flow** Instead of using the Wasserstein geometry as in [51], a few recent algorithms propose using gradient flow in the *Fisher–Rao* geometry [76, 63]. These methods follow a log-geometric path $\log \pi_\tau := (1-\tau)\log\pi_0 + \tau\log\pi_1$, $\tau \in [0,1]$, which connects an initial $\pi_0$ and a target $\pi_1$. While such bridges between prior and posterior have been used in Bayesian inference for many years [37], schemes in [76, 63] implement either gradient-free and gradient-informed estimators of particle transport using kernel methods, and allow practitioners to choose a time integration scheme over $\tau$. The theory of Fisher–Rao gradient flows has also seen much recent development [19, 77, 17].

**Consensus-based methods** Consensus-based sampling (CBS) comprises a system of interacting particles designed to sample from an unnormalized density [16]; it builds on consensus-based optimization methods for gradient-free non-convex optimization [67] and relates to collective behavior modeling [54, 27, 75, 23, 14]. At each iteration, particles are driven toward a consensus point, a weighted average favoring lower objective values, while exploring the space through additive noise. CBS is amenable to theoretical study from the mean-field perspective [15, 49, 36, 57], with explicit convergence rates, though these guarantees are generally limited to unimodal densities [16, 63].

### 2.2 MMD–optimizing methods

MMD is widely used in numerical analysis to assess quadrature rules [46, 5, 64, 52]. Using the reproducing kernel $\kappa$ for the RKHS $\mathcal{H}$, the MMD is given by

$$\mathrm{MMD}^2(\mu,\nu) \coloneqq \mathbb{E}_{\mu,\mu}[\kappa(X,X')] - 2\mathbb{E}_{\mu,\nu}[\kappa(X,Y)] + \mathbb{E}_{\nu,\nu}[\kappa(Y,Y')]\,. \tag{4}$$

Unlike the KL divergence, MMD is well defined even for mutually singular measures, as when comparing a continuous distribution to a discrete empirical distribution. Indeed, for a quadrature rule expressed as a discrete measure $p(\boldsymbol{w},Y) \coloneqq \sum_{i=1}^M w_i\delta_{y_i}$, the worst-case integration error over the unit ball of the RKHS, given by

$$\sup_{\|f\|_{\mathcal{H}}\leq 1}\left|\int_{\mathcal{X}} f(x)\mathrm{d}\pi(x) - \sum_{i=1}^M w_i f(y_i)\right|,$$

coincides with $\mathrm{MMD}(\pi, p(\boldsymbol{w},Y))$. This correspondence has motivated minimizing $\mathrm{MMD}(\pi, p(\boldsymbol{w},Y))$ using different approaches, including ridge leverage score sampling [4], determinantal point processes [8, 7], continuous volume sampling [9], Fekete points [53], recombination algorithms [43, 44], and gradient flows [2, 38]. These methods rely on symmetric kernels and, unfortunately, assume a normalized density. Using (4) for density $\tilde{\pi} = Z_{\tilde{\pi}}\pi$, we see

$$\mathrm{MMD}^2(\tilde{\pi}, p(\boldsymbol{w},Y)) = C_{\tilde{\pi}} - 2Z_{\tilde{\pi}}\mathbb{E}_{\pi}\left[\sum_{i=1}^M w_i\kappa(X,y_i)\right] + \sum_{i,j=1}^M w_i w_j \kappa(y_i,y_j), \tag{5}$$

for some constant $C_{\tilde{\pi}}$ independent of $p$. This affine relationship of the squared MMD with $Z_{\tilde{\pi}}$ makes the former infeasible to directly estimate, let alone minimize, for a general, unnormalized target.

The kernel Stein discrepancy (KSD), proposed for assessing sample quality [60, 22, 41], is a kernelized semi-metric invariant to this normalization and is equivalent to the MMD associated with a $\pi$-induced Stein kernel. KSD minimization has been used in the design of quadrature rules [18, 56]. By construction, however, first-order minimization of the KSD requires second-order information from the target density $\pi$ [56], and thus it is often difficult or impossible to perform this minimization without resorting to greedy methods [18] or otherwise using methods that scale poorly in dimension.

## 3 Main results

As discussed above, the minimization of KL divergence (and related functionals) underlies many methods for sampling from unnormalized densities, in large part because the associated objectives admit gradient-based optimization schemes that are invariant to the unknown normalizing constant. In contrast, the minimization of MMD (excluding Stein kernels) is typically seen as infeasible here. We challenge this notion and show that MMD minimization is achievable for unnormalized densities.

Recalling the definition of $p(\boldsymbol{w},Y)$, consider the function $F^\pi : \mathbb{R}^M \times \mathcal{X}^M \to \mathbb{R}^+$ given by

$$F^\pi(\boldsymbol{w},Y) \coloneqq \frac{1}{2}\mathrm{MMD}^2\left(\pi, p(\boldsymbol{w},Y)\right). \tag{6}$$

Observe that any critical point $(\boldsymbol{w}^*,Y^*) \in \mathbb{R}^M \times \mathcal{X}^M$ of $F$ is defined via

$$\nabla_{\boldsymbol{w}} F^\pi(\boldsymbol{w}^*,Y^*) = 0, \quad \nabla_{y_i} F^\pi(\boldsymbol{w}^*,Y^*) = 0. \tag{7}$$

We use conditions (7), which are necessary for optimality, as the desiderata for our quadrature rule. Recalling the notation $v_0$ and $v_1$ from (2) and (3), introduce the squared exponential kernel $\kappa(x,y) = \exp\!\big(-\frac{1}{2}\sigma^{-2}\|x-y\|^2\big)$ to express (7) as

$$\boldsymbol{K}(Y^*)\boldsymbol{w}^* = \boldsymbol{v}_0(Y^*), \quad \boldsymbol{W}^*\big(\boldsymbol{K}(Y^*)\boldsymbol{W}^*\,Y^* - \boldsymbol{v}_1(Y^*)\big) = \boldsymbol{0}, \tag{8}$$

where $\boldsymbol{W}^* \coloneqq \mathrm{diag}(\boldsymbol{w}^*)$ is the diagonal weight matrix. For a given $Y$, the first equation is linear in $\boldsymbol{w}^*$; we thus let the function $\boldsymbol{w}^* : \mathcal{X}^M \to \mathbb{R}^M$ describe the weights for configuration $Y$ as

$$\boldsymbol{w}^*(Y) = \boldsymbol{K}(Y)^{-1}\boldsymbol{v}_0(Y), \tag{9}$$

and define $\boldsymbol{W}^*(Y) = \mathrm{diag}(\boldsymbol{w}^*(Y))$.

The second condition in (8) is more challenging. A strategy to satisfy the condition is to alternate between solving the linear system (9) for $\boldsymbol{w}^*$ and updating the configuration $Y$ through the iteration

$$Y^{(t+1)} = (1-\eta)Y^{(t)} + \eta\Psi_{\mathrm{MSIP}}(Y^{(t)}). \tag{10}$$

Here $\eta > 0$ is a step size, and the function $\Psi_{\mathrm{MSIP}} : \mathcal{X}^M \to \mathcal{X}^M$ is defined by

$$\Psi_{\mathrm{MSIP}}(Y) := \boldsymbol{W}^*(Y)^{-1}\boldsymbol{K}(Y)^{-1}\boldsymbol{v}_1(Y). \tag{11}$$

This algorithm, known as *mean shift interacting particles* (MSIP), was introduced in [11] to minimize the MMD with respect to a probability measure $\pi$. In particular, it was proven that the dynamics (10) correspond to a preconditioned gradient descent of the function $F_M^\pi : \mathcal{X}^M \to \mathbb{R}^+$ defined as

$$F_M^\pi(Y) = \inf_{\boldsymbol{w}\in\mathbb{R}^M} F^\pi(\boldsymbol{w}, Y)\,. \tag{12}$$

The implementation of MSIP, however, was previously restricted to the "data-driven" setting where $\pi$ is available as an empirical distribution, as this input enables direct evaluation of the functions $v_0$, $v_1$.

In this work, we overcome this restriction and adapt MSIP to the case of an unnormalized density $\tilde{\pi}$. Our approach builds on the observation that the MSIP map $\Psi_{\mathrm{MSIP}}$, unlike $F^\pi$ itself, is invariant to the normalizing constant. Unfortunately, this property alone does not suffice for a systematic implementation of MSIP in the unnormalized setting. Two more challenges must be addressed: numerical stability and reliance of the map $\Psi_{\mathrm{MSIP}}$ on the embeddings $v_0$ and $v_1$.

### 3.1 Regularized mean shift interacting particles

We first grapple with extending the definition of the MSIP map to arbitrary particle configurations $Y$. The problem of conditioning primarily stems from the explicit inversion of the kernel matrix $\boldsymbol{K}(Y)$. A classical remedy for such numerical issues is to replace $\boldsymbol{K}(Y)$ with its regularized counterpart $\boldsymbol{K}_\lambda(Y) := \boldsymbol{K}(Y) + \lambda\boldsymbol{I}_M$ for some $\lambda > 0$. We thus use a *regularized* MSIP map $\Psi_{\mathrm{MSIP},\lambda}$ defined as

$$\Psi_{\mathrm{MSIP},\lambda}(Y) := \boldsymbol{W}_\lambda^*(Y)^{-1}\boldsymbol{K}_\lambda(Y)^{-1}\boldsymbol{v}_1(Y) \tag{13}$$

where $\boldsymbol{W}_\lambda^*(Y) = \mathrm{diag}(\boldsymbol{w}_\lambda^*(Y))$, with

$$\boldsymbol{w}_\lambda^*(Y) = \boldsymbol{K}_\lambda(Y)^{-1}\boldsymbol{v}_0(Y), \tag{14}$$

which is a regularization of the optimal weights defined by (9). This regularized map provides significantly more numerical stability—the matrix $\boldsymbol{K}_\lambda(Y)$ is nonsingular for any configuration $Y$—but, crucially, still yields preconditioned gradient descent of a penalized MMD, as we show below. Define the penalized objective function $F_\lambda^\pi : \mathbb{R}^M \times \mathcal{X}^M \to \mathbb{R}^+$ as

$$F_\lambda^\pi(\boldsymbol{w}, Y) \coloneqq \frac{1}{2}\mathrm{MMD}\left(\pi, p(\boldsymbol{w}, Y)\right)^2 + \frac{\lambda}{2}\|\boldsymbol{w}\|^2 = F^\pi(\boldsymbol{w}, Y) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2, \tag{15}$$

and consider the function $F_{M,\lambda}^\pi : \mathcal{X}^M \to \mathbb{R}_+$,

$$F_{M,\lambda}^\pi(Y) \coloneqq \inf_{\boldsymbol{w}\in\mathbb{R}^M} F_\lambda^\pi(\boldsymbol{w}, Y). \tag{16}$$

**Theorem 3.1.** *For the squared exponential kernel, we have*

$$\nabla F_{M,\lambda}^\pi(Y) = \sigma^{-2}\boldsymbol{W}_\lambda^*(Y)\Big(\boldsymbol{K}_\lambda(Y)\boldsymbol{W}_\lambda^*(Y)Y - \boldsymbol{v}_1(Y)\Big), \tag{17}$$

*where* $\boldsymbol{K}_\lambda(Y) := \boldsymbol{K}(Y) + \lambda\boldsymbol{I}_M$*, and* $\boldsymbol{W}_\lambda^*(Y) = \mathrm{diag}(\boldsymbol{w}_\lambda^*(Y))$*, with* $\boldsymbol{w}_\lambda^*(Y)$ *from* (14)*.*

Due to Theorem 3.1, we can see that the regularized dynamics

$$Y^{(t+1)} = (1-\eta)Y^{(t)} + \eta\Psi_{\mathrm{MSIP},\lambda}(Y^{(t)}), \tag{18}$$

for nonzero weights $\hat{\boldsymbol{w}}_\lambda$ leads to

$$Y^{(t+1)} = Y^{(t)} - \eta P_\lambda(Y)^{-1}\nabla F_{M,\lambda}^\pi(Y), \tag{19}$$

$$\text{with preconditioner } P_\lambda(Y) := \sigma^{-2}\boldsymbol{W}_\lambda^*(Y)\boldsymbol{K}_\lambda(Y)\boldsymbol{W}_\lambda^*(Y). \tag{20}$$

In other words, $\lambda$-regularization mollifies the target functional while ensuring that $\boldsymbol{K}_\lambda$ is invertible.

### 3.2 Numerical implementations

Implementing the iteration (18) requires the kernel mean embedding $\boldsymbol{v}_0$ and the kernelized first moment $\boldsymbol{v}_1$ for any set of particles $Y$. Unlike the data-driven setting where $\pi$ is an empirical measure, these functions do not admit closed-form expressions. In what follows, we propose computationally tractable estimators when using the squared exponential kernel. First, note that

$$\sigma^2 \nabla \log v_0(y) = \frac{\sigma^2 \nabla v_0(y)}{v_0(y)} = \frac{v_1(y) - y v_0(y)}{v_0(y)} = \frac{v_1}{v_0}(y) - y. \tag{21}$$

Therefore, we can consider the primitive quantities to be the pair of functions $(v_0, v_1)$ or the pair $(\log v_0,\ \sigma^2 \nabla \log v_0)$ with some equivalence. Our estimators rely on the following observation:

$$v_0(y) = \int_{\mathbb{R}^d} \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \pi(x) \mathrm{d}x = \omega_{\sigma,d} \int_{\mathbb{R}^d} \pi(y + \sigma\xi)\, \mathrm{d}\rho(\xi), \tag{22}$$

where $\rho$ is the standard multivariate Gaussian distribution and $\omega_{\sigma,d} := \left(\sqrt{2\pi}\sigma\right)^d$. Similarly,

$$v_1(y) = \omega_{\sigma,d} \int (y + \sigma\xi) \pi(y + \sigma\xi)\, \mathrm{d}\rho(\xi) \tag{23}$$

$$= y v_0(y) + \sigma \omega_{\sigma,d} \int \xi \pi(y + \sigma\xi)\, \mathrm{d}\rho(\xi). \tag{24}$$

The problem of estimating $v_0$ and $v_1$ over the unnormalized density $\pi$ thus reduces to computing integrals with respect to a multivariate Gaussian distribution. We propose to estimate these integrals via an inner quadrature rule. Suppose $(u_q, \xi_q)_{q=1}^Q \subset \mathbb{R} \times \mathcal{X}$ satisfies

$$\int \varphi(\xi) \mathrm{d}\rho(\xi) \approx \sum_{q=1}^Q u_q \varphi(\xi_q), \tag{25}$$

where $\varphi$ is a real-valued function defined on $\mathcal{X}$. A first choice for this *inner* quadrature rule is Monte Carlo, which involves sampling $\xi_1, \ldots, \xi_Q \overset{\text{iid}}{\sim} \rho$ with weights $u_q = Q^{-1}$. A second choice uses spherical radial quadrature rules [10, 1]. A third option is a simple one-point quadrature rule, i.e., using $Q = 1$, with $u_1 = 1$ and $\xi_1 = 0$. Using the discretization in (25) for the integrand $\xi \mapsto \pi(y + \sigma\xi)$ is straightforward and yields an approximation of $v_0$, denoted by $\hat{v}_0$ and defined by

$$\hat{v}_0^Q(y) := \omega_{\sigma,d} \sum_{q=1}^Q u_q \pi(y + \sigma\xi_q). \tag{26}$$

Since $v_1$ has multiple representations (23)–(24), strategies for approximating $v_1$ are discussed below.

**Gradient-free estimator** Applying the quadrature (25) to the first integral representation of $v_1$ (23) leads to a gradient-free approximation,

$$\hat{v}_1^{\mathrm{gf}}(y) := \omega_{\sigma,d} \sum_{q=1}^Q u_q (y + \sigma\xi_q) \pi(y + \sigma\xi_q). \tag{27}$$

**Stein estimator** Using Stein's identity [73] simplifies the representation in (24) of $v_1$:

$$v_1(y) = y\, v_0(y) + \sigma^2 \omega_{\sigma,d} \int_{\mathbb{R}^d} \nabla \pi(y + \sigma\xi) \mathrm{d}\rho(\xi). \tag{28}$$

Applying the quadrature (25) to (28), we derive the Stein estimator:

$$\hat{v}_1^{SQ}(y) = y\, \hat{v}_0^Q(y) + \sigma^2 \omega_{\sigma,d} \sum_{q=1}^Q (u_q \pi(y + \sigma\xi_q))\, \nabla \log \pi(y + \sigma\xi_q), \tag{29}$$

noting that $\nabla \pi = \pi \nabla \log \pi$, where $\nabla \log \pi$ is often referred to as the (Stein) score. For the one-point inner quadrature rule, the Stein estimator reduces to

$$\hat{v}_0^1(y) = \omega_{\sigma,d} \pi(y), \quad \hat{v}_1^{\mathrm{S1}}(y) = \omega_{\sigma,d}\, \pi(y) \big(y + \sigma^2 \nabla \log \pi(y)\big). \tag{30}$$

In particular, we see $\sigma^2 \nabla \log \pi(y)$ corresponds to $\hat{v}_1^{\mathrm{S1}}(y)/\hat{v}_0^1(y) - y$; this closely echoes $\nabla \log v_0$ in (21). Moreover, the one-point estimators can recover a Gaussian deconvolution of $\pi$: see Proposition B.1 for details. For this reason, we consider $\hat{v}_0^1$ and $\hat{v}_0^{\mathrm{S1}}$ to be *Fredholm* estimators.

### 3.3 MSIP invariance to normalization

Now that we have shown strategies for practically implementing MSIP, the regularized MSIP dynamics can be implemented using the following iteration:

$$Y^{(t+1)} = (1-\eta)Y^{(t)} + \eta\hat{\Psi}_{\mathrm{MSIP},\lambda}(Y^{(t)}), \tag{31}$$

where $\hat{\Psi}_{\mathrm{MSIP},\lambda}$ is obtained by replacing $v_0$ and $v_1$ by $\hat{v}_0$ and $\hat{v}_1$ respectively in (13).

$$\hat{\Psi}_{\mathrm{MSIP},\lambda}(Y) = \boldsymbol{W}_\lambda(Y)^{-1}\boldsymbol{K}_\lambda(Y)^{-1}\hat{\boldsymbol{v}}_1(Y). \tag{32}$$

We detail this method in Algorithm 1. Next, we show that this iteration is invariant to the normalizing constant of a target density $\pi$.

**Proposition 3.2.** *Let $\Psi_{\mathrm{MSIP},\lambda}(\cdot\,;\pi)$ be the regularized MSIP map for density $\pi$. For any $Y \in \mathcal{X}^M$,*

$$\Psi_{\mathrm{MSIP},\lambda}(Y;\pi) = \Psi_{\mathrm{MSIP},\lambda}(Y;\ C\,\pi)\ \ \forall C > 0. \tag{33}$$

The invariance of $\hat{\Psi}_{\mathrm{MSIP},\lambda}$ to normalization reasons similarly to Proposition 3.2. In essence, this invariance comes from the preconditioner $P_\lambda$ in (20). While it may seem at first glance that division by the weights $\boldsymbol{w}_\lambda^*$ echoes self-normalization in, e.g., importance sampling or CBS, such parallels are largely superficial. Whereas the drift term for any particle $y_i$ in CBS is nearly affine in its assigned weight $w_i$,[2]the dynamics of $y_i$ in MSIP are *inversely related* to the kernel-optimal weight $\hat{w}_i(Y)$.

### 3.4 Related kernelized methods

With details of our method in hand, we contrast it with other recent kernel methods used in sampling and optimization. One line of work considers interacting particle methods for Bayesian neural networks. In these approaches, repulsion and diversity are enforced at the level of network predictions rather than directly on the weights; see, e.g., [24]. Understanding the precise relationship between such function-space interactions and parameter-space interacting particle systems such as MSIP lies outside the scope of the present work. Polarized CBS [13] enriches classical CBS with interaction mechanisms designed to improve exploration of multi-modal distributions. While polarized CBS shares certain similarities with our gradient-free MSIPGF, the two approaches arise from fundamentally different principles: the former remains centered around consensus-driven dynamics, whereas MSIPGF constructs particle interactions through kernelized approximations of nonlocal quantities such as $v_0$ and $v_1$. The theoretical analysis in [13] is primarily carried out in the large-$M$ regime. Drifting [26] is a recent kernelized approach for generative modeling that constructs a vector field from the discrepancy between mean-shift maps associated with the target samples and the particle distribution. In contrast to MSIP, drifting is designed for sample-based generative modeling rather than for sampling from unnormalized densities.

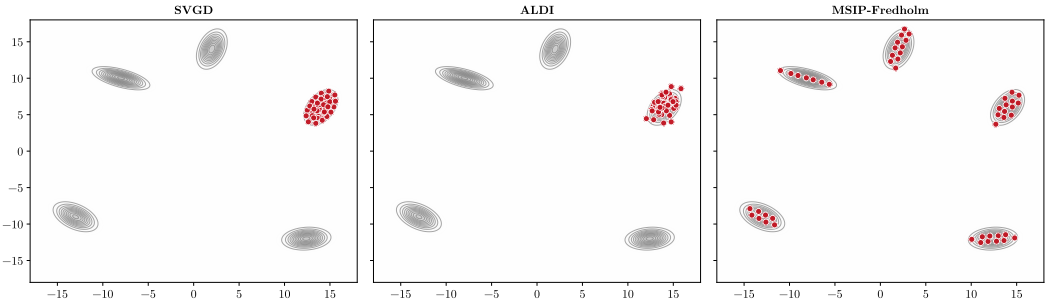


Figure 2: A comparison of our methods to other samplers on a mixture of five anisotropic Gaussians.

[2]The CBS drift is affine in the weights $\alpha w_i$ for $\alpha = 1/\sum w_i$. Specifically, the Jacobian of the drift w.r.t. $\boldsymbol{w}$ has $i$-th column $\alpha\boldsymbol{e}_i + \alpha^2\boldsymbol{w}$. This approaches an inverse relationship for one weight $w_i$ if $\alpha w_i \approx 1$, but remains nearly affine in all other weights.

# 4 Experiments

We now evaluate the proposed algorithms against established methods, including SVGD, ALDI, and CBS approaches. We use both synthetic distributions and real-world Bayesian posteriors. For error quantification, we normalize any weights to ensure unit sum.[3]

## 4.1 Mixtures of Gaussians

We use Gaussian mixtures to assess each method's ability to capture multi-modality, anisotropy and separation between modes. Further, for mixtures of Gaussians, one can evaluate the MMD analytically, which we use to assess the ability of the proposed algorithm in MMD minimization.

In the first simulation, we evaluate the proposed methods on a multi-modal target distribution defined by a five-component anisotropic Gaussian mixture in $\mathbb{R}^2$. We compare SVGD and gradient-informed ALDI to MSIP-Fredholm. For each number of particles $M$, we run multiple independent initializations from a Gaussian $\mathcal{N}(\mu, \mathbb{I}_2)$ and evolve the particles for a fixed number of iterations $T$, where $\mu = (18, 18)$. Figure 2 shows the configuration corresponding to the last iteration of each of the three algorithms. We observe that MSIP-Fredholm captures all modes of the distribution, while SVGD and ALDI suffer from mode collapse. This qualitative observation is validated by the quantitative comparison given in Figure 3, where the MMD at the last iteration of each algorithm is shown as a function of the particle count. We see MSIP outperforms SVGD and ALDI by a significant margin, especially when the number of particles is large.

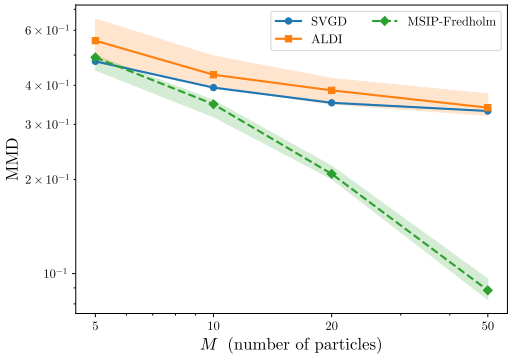


Figure 3: MMD decay with respect to the number of particles $M$ for three algorithms, over 20 independent trials. The shaded regions represent the 90% confidence intervals over the trials.

## 4.2 Benchmark sampling problems

In the second batch of experiments, we evaluate MSIP on a broader collection of target distributions. In particular, we compare several implementations of MSIP with state-of-the-art algorithms such as SVGD, ALDI, and CBS. More precisely, we consider three gradient-informed implementations defined in Section 3.2: i) MSIP-F, which is based on the Fredholm estimators (30); ii) MSIP GI-1, which is based on the Stein estimator, where the inner quadrature rule (25) corresponds to Monte Carlo sampling from the Gaussian distribution $\mathcal{N}(0, \sigma\mathbb{I}_d)$ with $Q = 1$; iii) MSIP-GI-10, which is also based on the Stein estimator, with the same inner quadrature rule but using $Q = 10$. We also consider a gradient-free implementation defined in Section 3.2: MSIP-GF, which is based on the estimator $\hat{v}_1^{\mathrm{gf}}$, where the inner quadrature rule (25) corresponds to Monte Carlo sampling from the Gaussian distribution $\mathcal{N}(0, \sigma\mathbb{I}_d)$ with $Q = 10$. Table 1 reports the resulting kernel Stein discrepancy (KSD) values on a collection of synthetic distributions with varying anisotropies and multi-modal behavior.

We observe that the different variants of MSIP consistently achieve either the lowest KSD values or values extremely close to the best-performing method across essentially all benchmark distributions. In contrast, competing approaches exhibit substantially less robustness across geometries and dimensions. In particular, SVGD frequently underperforms relative to MSIP variants, not only on

[3] All code used to generate the experiments is available at `https://github.com/Nodes-and-Kernels/nak_torch`.

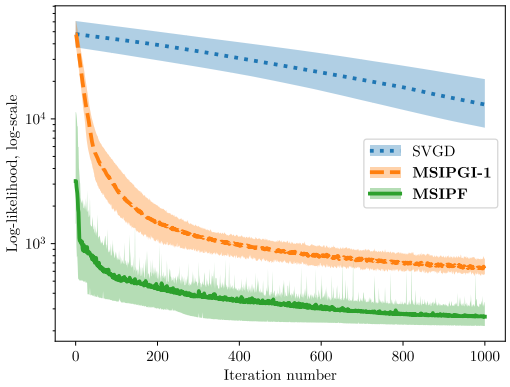


Figure 4: Log-likelihood of the neural network ensemble in the TwoMoons example over 20 independent trials. The shaded regions represent the 90% confidence interval as a function of the iteration number.

Table 1: Error of different algorithms for synthetic distributions using KSD with an inverse multiquadric kernel. We report mean and standard deviation over many trials; the smallest row-wise mean is bolded. Empty cells correspond to an algorithm not converging.

| | | Gradient-informed algorithms | | | | | Gradient-free algorithms | |
|---|---|---|---|---|---|---|---|---|
| | Dim. | **MSIPF** | **MSIPGI-1** | **MSIPGI-10** | SVGD | ALDI | **MSIPGF** | CBS |
| GMM2 | 2 | 1.000 (0.056) | 1.489 (0.499) | 3.126 (5.286) | 1.223 (0.039) | **0.930 (0.094)** | 6.014 (5.028) | 1.103 (0.090) |
| GMM5 | 5 | 1.510 (0.122) | 3.495 (4.684) | **1.375 (0.022)** | 1.499 (0.004) | 1.516 (0.054) | 6.916 (12.33) | 3.293 (0.336) |
| GMM10 | 10 | 2.229 (0.002) | **1.970 (0.010)** | 2.041 (0.039) | 2.172 (0.002) | 2.185 (0.039) | 3.777 (2.145) | 3.899 (0.369) |
| GMM20 | 20 | 3.395 (1.531e-05) | **3.012 (0.030)** | 3.014 (0.040) | 3.060 (0.003) | 3.355 (0.171) | 3.110 (0.074) | 5.061 (0.345) |
| Joker | 2 | 10.26 (0.474) | **8.493 (0.016)** | 8.563 (0.068) | 11.11 (1.579) | 9.286 (0.185) | 8.732 (0.369) | 10.02 (1.475) |
| Funnel2 | 2 | 9.104 (0.085) | **8.618 (0.022)** | 8.620 (0.017) | 9.073 (0.145) | 9.008 (0.091) | 8.763 (0.019) | 95.40 (176.8) |
| Funnel5 | 5 | 30.44 (5.118) | **14.25 (0.021)** | 14.31 (0.029) | 18.64 (2.574) | 14.43 (0.210) | 14.30 (0.021) | 14.98 (0.615) |
| Funnel10 | 10 | 83.79 (17.37) | **20.24 (0.039)** | 20.48 (0.033) | 24.06 (1.593) | 20.34 (0.167) | 20.40 (0.075) | 21.86 (1.396) |
| Funnel20 | 20 | 131.0 (16.90) | 28.92 (0.160) | 29.97 (0.408) | 67.36 (22.99) | **28.66 (0.122)** | 29.86 (0.377) | 34.05 (3.475) |
| Funnel50 | 50 | 236.4 (0.617) | **46.68 (0.687)** | 49.75 (1.725) | 242.5 (142.7) | — | 47.64 (0.887) | 58.30 (10.52) |
| Himmelblau | 2 | 8.129 (0.182) | 9.778 (1.952) | **7.671 (0.188)** | 14.24 (0.311) | 14.33 (2.558) | 8.071 (0.335) | — |

challenging targets such as funnel, joker, and Himmelblau distributions, but also on several Gaussian mixture models, especially when the evaluation bandwidth matches the characteristic scale used to construct the targets. These observations also highlight a limitation of KSD-based evaluation: on multi-modal distributions, small KSD values do not necessarily imply accurate global exploration or faithful mode allocation. This phenomenon appears sharply when examining visualizations of the two-dimensional distributions, as in Section B.4. Similarly, ALDI performs strongly on funnel distributions, where its affine-invariant structure is advantageous, but is generally less competitive on the remaining targets. By contrast, the gradient-free variant MSIP-GF remains remarkably competitive across nearly all synthetic experiments, including difficult high-dimensional settings. Overall, the MSIP family is the only class of methods in this benchmark that consistently maintains strong and stable performance simultaneously across multi-modal, anisotropic, heavy-tailed, and high-dimensional targets.

Table 2 presents a broad evaluation of our methods on non-synthetic target distributions, where any large deviation suggests near-failure in a given trial.[4] Remarkably, virtually all other methods often fail without careful tuning of step size across problems; the ALDI results, in particular, show the peril of algorithmic sensitivity to step size. As a case study, Figure 4 demonstrates the convergence in log-likelihood of MSIP for the Bayesian neural network example of classification in a problem with $d = 201$ parameters; we see that both MSIP-Fredholm and MSIP-GI converge much quicker than the SVGD results, showing the strength of our preconditioned dynamics.

Note that the costs of MSIP-Fredholm and the one-point MSIP GI-1 are both *equivalent* in number of density evaluations to SVGD. On the other hand, the inner quadrature rules used for MSIP GI-10 and MSIP GF both require more information from the target $\pi$. The non-local information these

[4]For each (weighted) point set and unnormalized target density $\widetilde{\pi}$, we calculate the log-likelihood as $\mathcal{L}(Y, \boldsymbol{w}; \widetilde{\pi}) := -\sum_{k=1}^{M} w_k \log \widetilde{\pi}(y_k)$; this can be negative for unnormalized $\widetilde{\pi}$.

Table 2: Error on benchmark problems in log-likelihood (LL) and kernel Stein discrepancy (KSD). Examples are suffixed by the dimension $d$ of the target. An empty cell denotes failure of the algorithm.

| | | **MSIPF** | **MSIPGI-1** | SVGD | ALDI | **MSIPGF** | CBS |
|---|---|---|---|---|---|---|---|
| LL | Schools10 | **6.553 (1.63)** | 12.76 (1.04) | 11.89 (2.67) | 9.152 (0.800) | 21.68 (11.2) | 9.823 (2.43) |
| | COVID51 | **4616 (409)** | 6782 (155) | 471239 (37239) | — | 780936 (42414) | 655235 (68207) |
| | CovType56 | 293827 (68258) | 645767 (59017) | **287376 (1911)** | — | 332276 (11689) | — |
| | ABPDE64 | **546 (140)** | 1062 (563) | 3203 (4292) | — | 5447 (6977) | — |
| | TwoMoons201 | **258.60 (27.07)** | 638.76 (50.87) | 13653 (3594) | — | 45670 (6570) | — |
| KSD | Schools10 | 28.7 (7.5) | 9.6 (0.18) | 9.50 (1.1) | **9.04 (0.04)** | 20.2 (18) | 10.6 (1.8) |
| | COVID51 | 507 (74) | **106 (1.7)** | 37817 (1028) | — | 42176 (850) | 45396 (1393) |
| | CovType56 | 35140 (21703) | 15373 (1216) | **1865.2 (433)** | — | 3667 (1235) | — |
| | ABPDE64 | 164 (20) | **42.3 (2.3)** | 818 (1782) | — | 4293 (9609) | — |
| | TwoMoons201 | 267 (50) | **168 (15)** | 449 (76) | — | 801 (45) | — |

estimators glean comes from a user-specified budget; the corresponding evaluations, however, can be performed in parallel. Indeed, vectorization and mapping operations common in machine learning toolkits (e.g., the `vmap` transformation in torch/JAX) yield virtually identical cost to MSIP-Fredholm for small quadrature rules, depending on the problem dimension $d$ and the number of particles $M$.

In Section B.4, we report additional two-dimensional case studies comparing results of our new algorithms with those produced by several other methods, at comparable computational cost. In these examples, MSIP methods produce consistently high-quality quadrature rules, with well-spaced point sets that capture all essential features of the target distributions.

## 5 Conclusion

We have shown that MSIP is an effective new approach for producing *weighted quadratures* of distributions known only through their unnormalized density. To our knowledge, it is the most scalable method for MMD minimization in this setting. Our experiments demonstrate that MSIP algorithms enjoy robust performance over a very broad range of target distributions, achieving approximation quality that is consistently competitive with—and often superior to—current state-of-the-art sampling approaches. A key strength of MSIP is that its dynamics balance systematic modal coverage (i.e., quickly finding all modes of a target) with accurate representation of the shape and extent of each mode.

Underlying MSIP is the internal quadrature rule, a distinctive degree of freedom not afforded in typical approaches. Through the quantities $v_0$ and $v_1$, this quadrature highlights non-local information, in lieu of dynamics driven by the score function alone. Our experiments suggest that accurately approximating these global kernel-based quantities can lead to substantial improvements in performance, particularly for challenging multi-modal and anisotropic targets.

Future work will focus on establishing theoretical guarantees for MSIP, including convergence and approximation results, as the nonconvex optimization at the core of the method remains challenging analytically. We also plan to advance systematic and adaptive strategies for selecting the internal quadrature rule and associated kernel hyperparameters.

**Acknowledgments** DS acknowledges support from a 2025–2026 MathWorks Fellowship. AB and YMM acknowledge support from the ExxonMobil Technology and Engineering Company. DS and YMM acknowledge support from the US Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research (ASCR), via the M2dt MMICC center under award number DE-SC0023187 and through the Scientific Discovery through Advanced Computing (SciDAC) FASTMath Institute, under contract number DE-AC52-07NA27344. AB and YMM also acknowledge support from DOE ASCR under award numbers DE-SC0023188 and DE-SC0026245.

The authors declare no additional competing interests.

## References


[1] I. Arasaratnam and S. Haykin. Cubature Kalman Filters. *IEEE Transactions on Automatic Control*, 54(6):1254–1269, June 2009. ISSN 0018-9286, 1558-2523. doi: 10.1109/TAC.2009.2019800.

[2] M. Arbel, A. Korba, A. Salim, and A. Gretton. Maximum mean discrepancy gradient flow. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] D. Aristoff and W. Bangerth. A Benchmark for the Bayesian Inversion of Coefficients in Partial Differential Equations. *SIAM Review*, 65(4):1074–1105, November 2023. ISSN 0036-1445, 1095-7200.

[4] F. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *The Journal of Machine Learning Research*, 18(1):714–751, 2017.

[5] F. Bach, S. Lacoste-Julien, and G. Obozinski. On the equivalence between herding and conditional gradient algorithms. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 1355–1362, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

[6] S. Banerjee, K. Balasubramanian, and P. Ghosal. Improved finite-particle convergence rates for stein variational gradient descent. In *The Thirteenth International Conference on Learning Representations*, 2025.

[7] A. Belhadji. An analysis of Ermakov–Zolotukhin quadrature using kernels. *Advances in Neural Information Processing Systems*, 34:27278–27289, 2021.

[8] A. Belhadji, R. Bardenet, and P. Chainais. Kernel quadrature with DPPs. *Advances in Neural Information Processing Systems*, 32:12907–12917, 2019.

[9] A. Belhadji, R. Bardenet, and P. Chainais. Kernel interpolation with continuous volume sampling. *Proceedings of the 37th International Conference on Machine Learning*, pages 725–735, 2020.

[10] A. Belhadji, Q. J. Zhu, and Y. Marzouk. On the design of scalable, high-precision spherical-radial fourier features. *arXiv preprint arXiv:2408.13231*, 2024.

[11] A. Belhadji, D. Sharp, and Y. M. Marzouk. Weighted quantization using MMD: From mean field to mean shift using gradient flows. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026.

[12] J. Besag. Comments on "Representations of knowledge in complex systems" by U. Grenander and M. I. Miller. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(4): 591–592, 1994.

[13] L. Bungert, T. Roith, and P. Wacker. Polarized consensus-based dynamics for optimization and sampling. *Mathematical Programming*, 211(1-2):125–155, May 2025. ISSN 0025-5610, 1436-4646.

[14] J. A. Carrillo, M. Fornasier, G. Toscani, and F. Vecil. Particle, kinetic, and hydrodynamic models of swarming. In *Mathematical modeling of collective behavior in socio-economic and life sciences*, pages 297–336. Springer, 2010.

[15] J. A. Carrillo, Y-P. Choi, C. Totzeck, and O. Tse. An analytical framework for consensus-based global optimization method. *Mathematical Models and Methods in Applied Sciences*, 28(06): 1037–1066, 2018.

[16] J. A. Carrillo, F. Hoffmann, A. M. Stuart, and U. Vaes. Consensus-based sampling. *Studies in Applied Mathematics*, 148(3):1069–1140, 2022.

[17] J. A. Carrillo, Y. Chen, D. Z. Huang, J. Huang, and D. Wei. Fisher–Rao gradient flow: Geodesic convexity and functional inequalities. *SIAM Journal on Mathematical Analysis*, 58(2):1062–1099, 2026.

[18] W.Y. Chen, L. Mackey, J. Gorham, F.–X. Briol, and C. Oates. Stein points. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 844–853. PMLR, 10–15 Jul 2018.

[19] Y. Chen, D. Z. Huang, J. Huang, S. Reich, and A. M. Stuart. Gradient flows for sampling: Mean-field models, Gaussian approximations and affine invariance. *arXiv preprint arXiv:2302.11024*, 2023.

[20] S. Chewi, T. Le Gouic, C. Lu, T. Maunu, and P. Rigollet. SVGD as a kernelized Wasserstein gradient flow of the chi-squared divergence. *Advances in Neural Information Processing Systems*, 33:2098–2109, 2020.

[21] S. Chewi, J. Niles-Weed, and P. Rigollet. *Statistical Optimal Transport: École d'Été de Probabilités de Saint-Flour XLIX – 2019*, volume 2364 of *Lecture Notes in Mathematics*. Springer Nature Switzerland, Cham, 2025. ISBN 978-3-031-85159-9 978-3-031-85160-5.

[22] K. Chwialkowski, H. Strathmann, and A. Gretton. A kernel test of goodness of fit. In *International conference on machine learning*, pages 2606–2615. PMLR, 2016.

[23] F. Cucker and S. Smale. On the mathematics of emergence. *Japanese Journal of Mathematics*, 2(1):197–227, 2007.

[24] F. D'Angelo and V. Fortuin. Repulsive deep ensembles are Bayesian. *Advances in Neural Information Processing Systems*, 34:3451–3465, 2021.

[25] A. Das and D. Nagaraj. Provably fast finite particle variants of SVGD via virtual particle stochastic approximation. *Advances in Neural Information Processing Systems*, 36:49748–49760, 2023.

[26] M. Deng, H. Li, T. Li, Y. Du, and K. He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.

[27] M. Dorigo and C. Blum. Ant colony optimization theory: A survey. *Theoretical computer science*, 344(2-3):243–278, 2005.

[28] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987. doi: 10.1016/0370-2693(87)91197-X.

[29] A. Duncan, N. Nüsken, and L. Szpruch. On the geometry of Stein variational gradient descent. *Journal of Machine Learning Research*, 24(56):1–39, 2023.

[30] G. Evensen. The ensemble kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53(4):343–367, 2003.

[31] S. Flaxman, S. Mishra, A. Gandy, H. Unwin, H. Coupland, T. Mellan, H. Zhu, T. Berah, J. Eaton, P. Perez-Guzman, et al. Report 13: Estimating the number of infections and the impact of non-pharmaceutical interventions on COVID-19 in 11 European countries. Technical report, Imperial College London, 2020.

[32] A. Garbuno-Inigo, F. Hoffmann, W. Li, and A. M. Stuart. Interacting Langevin diffusions: Gradient structure and ensemble Kalman sampler. *SIAM Journal on Applied Dynamical Systems*, 19(1):412–441, 2020.

[33] A. Garbuno-Inigo, N. Nüsken, and S. Reich. Affine invariant interacting Langevin dynamics for Bayesian inference. *SIAM Journal on Applied Dynamical Systems*, 19(3):1633–1658, 2020.

[34] W. Gautschi. *A Survey of Gauss-Christoffel Quadrature Formulae*. Birkhauser Verlag, 1981.

[35] A. Gelman and J. Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press, 2006.

[36] N. J. Gerber, F. Hoffmann, and U. Vaes. Mean-field limits for Consensus-Based Optimization and Sampling. *ESAIM: Control, Optimisation and Calculus of Variations*, 31:74, 2025. ISSN 1292-8119, 1262-3377.

[37] C. J. Geyer and E. A. Thompson. Annealing Markov Chain Monte Carlo with Applications to Ancestral Inference. *Journal of the American Statistical Association*, September 1995.

[38] E. Gladin, P. Dvurechensky, A. Mielke, and J.-J. Zhu. Interaction-force transport gradient flows. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[39] G. H. Golub and J. H. Welsch. Calculation of Gauss quadrature rules. *Mathematics of computation*, 23(106):221–230, 1969.

[40] J. Goodman and J. Weare. Ensemble samplers with affine invariance. *Communications in Applied Mathematics and Computational Science*, 5(1):65–80, 2010. doi: 10.2140/camcos.2010.5.65.

[41] J. Gorham and L. Mackey. Measuring sample quality with kernels. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1292–1301. PMLR, August 2017.

[42] H. Haario, M. Laine, A. Mira, and E. Saksman. DRAM: Efficient adaptive MCMC. *Statistics and Computing*, 16(4):339–354, 2006. doi: 10.1007/s11222-006-9438-0.

[43] S. Hayakawa, H. Oberhauser, and T. Lyons. Positively weighted kernel quadrature via subsampling. *Advances in Neural Information Processing Systems*, 35:6886–6900, 2022.

[44] S. Hayakawa, H. Oberhauser, and T. Lyons. Sampling-based Nyström approximation and kernel quadrature. In *International Conference on Machine Learning*, pages 12678–12699. PMLR, 2023.

[45] Y. He, K. Balasubramanian, B. K. Sriperumbudur, and J. Lu. Regularized Stein variational gradient flow. *Foundations of Computational Mathematics*, 25(4):1199–1257, 2025.

[46] F. Hickernell. A generalized discrepancy and quadrature error bound. *Mathematics of computation*, 67(221):299–322, 1998.

[47] D. M. Himmelblau. *Applied Nonlinear Programming*. McGraw-Hill, New York, 1972. ISBN 978-0-07-028921-5.

[48] M. D. Hoffman and A. Gelman. The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[49] H. Huang, J. Qiu, and K. Riedl. On the global convergence of particle swarm optimization methods. *Applied Mathematics & Optimization*, 88(2):30, 2023.

[50] F. Igea and A. Cicirello. Cyclical Variational Bayes Monte Carlo for efficient multi-modal posterior distributions evaluation. *Mechanical Systems and Signal Processing*, 186:109868, March 2023. ISSN 08883270.

[51] R. Jordan, D. Kinderlehrer, and F. Otto. The variational formulation of the Fokker–Planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

[52] M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

[53] T. Karvonen, S. Särkkä, and K. Tanaka. Kernel-based interpolation at approximate Fekete points. *Numerical Algorithms*, 87:445–468, 2021.

[54] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.

[55] A. Korba, A. Salim, M. Arbel, G. Luise, and A. Gretton. A non-asymptotic analysis for Stein variational gradient descent. *Advances in Neural Information Processing Systems*, 33: 4672–4682, 2020.

[56] A. Korba, P.C. Aubin-Frankowski, S. Majewski, and P. Ablin. Kernel Stein discrepancy descent. In *International Conference on Machine Learning*, pages 5719–5730. PMLR, 2021.

[57] M. Koß, S. Weissmann, and J. Zech. On the Mean-Field Limit of Consensus-Based Methods. *Mathematical Methods in the Applied Sciences*, page mma.70343, November 2025. ISSN 0170-4214, 1099-1476.

[58] Q. Liu. Stein variational gradient descent as gradient flow. *Advances in neural information processing systems*, 30, 2017.

[59] Q. Liu and D. Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.

[60] Q. Liu, J. Lee, and M. Jordan. A kernelized Stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, pages 276–284. PMLR, 2016.

[61] T. Liu, P. Ghosal, K. Balasubramanian, and N. Pillai. Towards understanding the dynamics of Gaussian-Steinvariational gradient descent. *Advances in Neural Information Processing Systems*, 36:61234–61291, 2023.

[62] M. Magnusson, P. Bürkner, and A. Vehtari. posteriordb: a set of posteriors for Bayesian inference and probabilistic programming, October 2023.

[63] A. Maurais and Y. Marzouk. Sampling in unit time with kernel Fisher–Rao flow. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 35138–35162. PMLR, 21–27 Jul 2024.

[64] K. Muandet, K. Fukumizu, B. Sriperumbudur, B. Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2): 1–141, 2017.

[65] R. M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, Berlin, Heidelberg, 1996. ISBN 0387947248.

[66] R. M. Neal. Slice sampling. *The Annals of Statistics*, 31(3), June 2003. ISSN 0090-5364. doi: 10.1214/aos/1056562461.

[67] R. Pinnau, C. Totzeck, O. Tse, and S. Martin. A consensus-based model for global optimization and its mean-field limit. *Mathematical Models and Methods in Applied Sciences*, 27(01): 183–204, 2017.

[68] C. P. Robert, G. Casella, and G. Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

[69] G. O. Roberts and R. L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996.

[70] A. Salim, L. Sun, and P. Richtarik. A convergence theory for SVGD in the population limit under Talagrand's inequality T1. In *International Conference on Machine Learning*, pages 19139–19152. PMLR, 2022.

[71] F. Santambrogio. {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of Mathematical Sciences*, 7:87–154, 2017.

[72] J. Shi and L. Mackey. A finite-particle convergence rate for Stein variational gradient descent. *Advances in Neural Information Processing Systems*, 36:26831–26844, 2023.

[73] C. M. Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981.

[74] L. Sun, A. Karagulyan, and P. Richtarik. Convergence of Stein variational gradient descent under a weaker smoothness condition. In *International Conference on Artificial Intelligence and Statistics*, pages 3693–3717. PMLR, 2023.

[75] G. Toscani. Kinetic models of opinion formation. *Communications in Mathematical Sciences*, 4(3):481–496, 2006.

[76] L. Wang and N. Nüsken. Measure transport with kernel mean embeddings. *arXiv preprint arXiv:2401.12967*, 2024.

[77] J.-J. Zhu and A. Mielke. Kernel approximation of Fisher–Rao gradient flows. *arXiv preprint*, 2024.

[78] J. Zhuo, C. Liu, J. Shi, J. Zhu, N. Chen, and B. Zhang. Message passing stein variational gradient descent. In *International Conference on Machine Learning*, pages 6018–6027. PMLR, 2018.

# A Numerics

## A.1 The algorithm

**Algorithm 1:** Regularized MSIP

**Input:** Density $\pi$, step size $\eta$, number of iterations $T$, regularization parameter $\lambda$, hybridization rate $\gamma$, inner quadrature rule $(u_q, \xi_q)_{q=1}^Q \subset \mathbb{R} \times \mathcal{X}$, initialization $Y^{(0)} = \{y_1^{(0)}, \dots, y_M^{(0)}\}$

**Output:** Last configuration $Y^{(T)}$

**for** $t = 0, 1, \dots, T-1$ **do**
  **for** $i = 1, \dots, M$ **do**
    $\hat{v}_0(y_i) \leftarrow \omega_{\sigma,d} \sum_{q=1}^Q u_q \pi(y_i + \sigma\xi_q)$ ;
    $\hat{v}_1^H(y_i) \leftarrow (1-\gamma)\hat{v}_1^{\mathrm{gf}}(y) + \gamma \hat{v}_1^S(y)$;
  **end**
  $\boldsymbol{K}_\lambda(Y) \leftarrow \boldsymbol{K}(Y) + \lambda \boldsymbol{I}_M$ ;
  $\boldsymbol{w}_\lambda^*(Y) \leftarrow \boldsymbol{K}_\lambda(Y)^{-1}\hat{\boldsymbol{v}}_0(Y)$;
  $\boldsymbol{W}_\lambda^*(Y) \leftarrow \mathrm{diag}(\boldsymbol{w}_\lambda^*(Y))$ ;
  $\Psi^{(t)} \leftarrow (\boldsymbol{K}_\lambda(Y)\boldsymbol{W}_\lambda^*(Y))^{-1}\hat{\boldsymbol{v}}_1^H(Y)$;
  $Y^{(t+1)} \leftarrow (1-\eta)\, Y^{(t)} + \eta \Psi^{(t)}$;
**end**

## A.2 MSIP for mixtures of Gaussians

Consider a mixture of Gaussians

$$\pi(x) = \sum_{k=1}^K m_k \mathcal{N}(x;\, \mu_k,\, \Sigma_k) \tag{34}$$

with

$$\mathcal{N}(x;\, \mu_k, \Sigma_k) := \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right) \tag{35}$$

We consider the kernel $\kappa_\sigma(x,y) = \exp(-\frac{1}{2\sigma^2}\|x-y\|^2) = Z_\sigma \mathcal{N}(x;\, y, \sigma^2 I)$ for $Z_\sigma = (2\pi\sigma^2)^{d/2}$, and we obtain

$$v_0(y) = \int \kappa_\sigma(x,y)\pi(x)\mathrm{d}x = Z_\sigma \sum_{k=1}^K m_k \mathcal{N}(y \mid \mu_k, \widetilde{\Sigma}_k) \tag{36}$$

$$= \frac{Z_\sigma}{(2\pi)^{d/2}} \sum_{k=1}^K \frac{m_k}{|\widetilde{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(y-\mu_k)^T\widetilde{\Sigma}_k^{-1}(y-\mu_k)\right) \tag{37}$$

where $\widetilde{\Sigma}_k = \Sigma_k + \sigma^2 I$. Now, let $\widetilde{\Sigma}_k = L_k L_k^\top$ be the Cholesky decomposition of $\widetilde{\Sigma}_k$ and let $z_k := L_k^{-1}(y-\mu_k)$. We have $\log|\widetilde{\Sigma}_k|^{1/2} = \sum_{j=1}^d \log[L_k]_{jj}$ and, thus,

$$\log v_0(y) = \log \sum_{k=1}^K \exp\left(\log m_k - \sum_{j=1}^d \log[L_k]_{jj} - \frac{1}{2}\|z_k\|_2^2\right) + d\log\sigma \tag{38}$$

Therefore,

$$\nabla \log v_0(y) = \sum_{k=1}^K r_k(y)\left(-\widetilde{\Sigma}_k^{-1}(y-\mu_k)\right), \tag{39}$$

where

$$r_k(y) := \frac{m_k\, \mathcal{N}(y;\, \mu_k,\, \widetilde{\Sigma}_k)}{\sum_{i=1}^K m_i\, \mathcal{N}(y;\, \mu_i,\, \widetilde{\Sigma}_i)} \tag{40}$$

Using the Cholesky factors, $\widetilde{\Sigma}_k^{-1}(y-\mu_k) = L_k^{-\top} z_k$, we get

$$\nabla \log v_0(y) = -\sum_{k=1}^{K} r_k(y) L_k^{-\top} z_k. \tag{41}$$

This last identity is purely for computational convenience.

### A.3 Experimental details

We now discuss the details of the experiments provided in Section 4.2. All experiments were performed on standard consumer-grade CPUs, except for ABPDE, which was performed using an RTX4090. All code was implemented in torch.

**GMM** We consider five-component Gaussian mixture models. For $d \geq 5$, GMM-$d$ is the equally weighted mixture with component means $\mu_k = 7.5e_k$ and covariances $\Sigma_k = 0.5I_d$, where $e_k \in \mathbb{R}^d$ is the $k$th standard basis vector. For GMM2, we instead use a mixture of anisotropic Gaussians.

**Joker** The Joker distribution is a mixture of two Gaussian densities of different anisotropy and a "banana" distribution that is *not sub-Gaussian*—this is a preliminary test of robustness to non-Gaussianity. See Figure 8 for a visualization.

**Funnel** The funnel distributions, originally from [66], are classic test cases used in benchmarking sampling algorithms in the presence of high anisotropy; these have been used to benchmark methods in, e.g., [63]. Up to a constant, the density is expressed as:

$$\pi(x_1, x_{2:d}) \propto \mathcal{N}(x_1; \mu, \sigma^2 I)\mathcal{N}\big(x_{2:d}; 0, \exp(x_1)I\big).$$

**Himmelblau** Similarly, the Himmelblau function [47], with unnormalized density

$$\pi(x) \propto \exp\left(-(x_1^2 + x_2 - 11)^2 - (x_1 + x_2^2 - 7)^2\right)$$

is a two-dimensional test case common for checking the behavior of sampling methods [50].

**Schools** We use the "eight schools" posterior, implemented in `posteriordb` [62], as described in [35].

**COVID** Similar to the above, we use a posterior from a report describing the spread and contagion of COVID-19 [31], as implemented in `posteriordb`.

**CovType** We perform hierarchical Bayesian logistic regression on a large dataset with an identical setup to [59]. That is, mini-batching 50 random data samples out of the entire dataset at each step of the algorithm.

**ABPDE** We use a PDE-constrained inverse problem intended as a benchmark for Bayesian algorithms, as proposed by Aristoff and Bangerth [3]. In brief, the Bayesian problem here is inference of piecewise-constant material properties on a two-dimensional square domain, in a linear elasticity PDE. The map from material properties to the PDE solution (strain/displacement) is nonlinear, so the posterior is intrinsically non-Gaussian. Non-Gaussianity is amplified by treating the material properties on a logarithmic scale (to enforce positivity). To avoid an inverse crime, we add Gaussian noise of the appropriate amount to the proposed optimum.

**TwoMoons** We consider the optimization of Bayesian neural networks for a two-dimensional classification task. The network is a simple multilayer perception with width 50 and one hidden layer. We use mini batches of size 64 at each step of each algorithm.

We provide several cross-algorithm hyperparameters in Table 3. For each example, we take 25 particles in every method, CBS uses inverse temperature $\beta = 0.9$, and we use ten Monte Carlo samples for the inner quadrature rule of MSIPGF as well as MSIPGI-10.

Table 3: Algorithm hyperparameters

| Example | Dimension | Step Size | MSIP Bandwidth | KSD Bandwidth | Steps | Trials | Bounds |
|---|---|---|---|---|---|---|---|
| GMM5 | 5 | 0.5 | 0.5 | 0.5 | 1000 | 10 | $(-10^3, 10^3)$ |
| GMM10 | 10 | 0.5 | 0.5 | 0.5 | 1000 | 10 | $(-10^3, 10^3)$ |
| GMM20 | 20 | 0.5 | 0.5 | 0.5 | 1000 | 10 | $(-10^3, 10^3)$ |
| Joker | 2 | 0.5 | 0.1 | 0.1 | 1000 | 10 | $(-10^3, 10^3)$ |
| Funnel2 | 2 | 0.5 | 0.1 | 0.1 | 1000 | 10 | $(-10^3, 10^3)$ |
| Funnel5 | 5 | 0.05 | 0.1 | 0.1 | 1000 | 10 | $(-10^3, 10^3)$ |
| Funnel10 | 10 | 0.05 | 0.1 | 0.1 | 1000 | 10 | $(-10^3, 10^3)$ |
| Funnel20 | 20 | 0.05 | 0.1 | 0.1 | 1000 | 10 | $(-10^3, 10^3)$ |
| Himmelblau | 2 | 0.1 | 0.05 | 0.1 | 1000 | 10 | $(-10^3, 10^3)$ |
| Schools | 10 | $10^{-3}$ | 0.1 | 0.1 | 1000 | 20 | $(-10^3, 10^3)$ |
| COVID | 51 | $10^{-3}$ | 0.1 | 0.1 | 500 | 10 | $(-10^3, 10^3)$ |
| CovType | 56 | $10^{-3}$ | 0.1 | 0.1 | 1000 | 20 | $(-10^3, 10^3)$ |
| ABPDE | 64 | $10^{-3}$ | 0.1 | 0.1 | 1000 | 20 | $(-10^3, 10^3)$ |
| TwoMoons | 201 | $10^{-3}$ | 0.1 | 0.1 | 1000 | 20 | $(-10^3, 10^3)$ |

**Sensitivity to the level of regularization** We investigate the effect of the level of regularization on the final quantization error of MSIP-Fredholm using the five-component anisotropic Gaussian mixture considered in the main text. We use the regularized kernel matrix $K_\lambda(Y) = K(Y) + \lambda I$, with $\lambda \in 10^{-10}, 10^{-9}, \ldots, 10^{0}$, and particle counts $M \in 50, 100, 250$. For each configuration, we perform 50 trials of 10,000 iterations, using a fixed step size $\eta = 0.2$ and Gaussian kernel bandwidth 0.1 for both MSIP and MMD evaluation. Particles are initialized independently from $\mathcal{N}((18, 18), I_2)$, with the same initial configurations reused across regularization levels for each particle count. Figure 5 reports the mean final MMD, evaluated using the normalized MSIP weights, with error bars indicating one standard deviation across trials. The MMD remains nearly constant over several orders of magnitude in the small-regularization regime, whereas stronger regularization increases the error, particularly for larger values of $M$. These results suggest that a broad range of small regularization level $\lambda$ yields comparable final MMD.

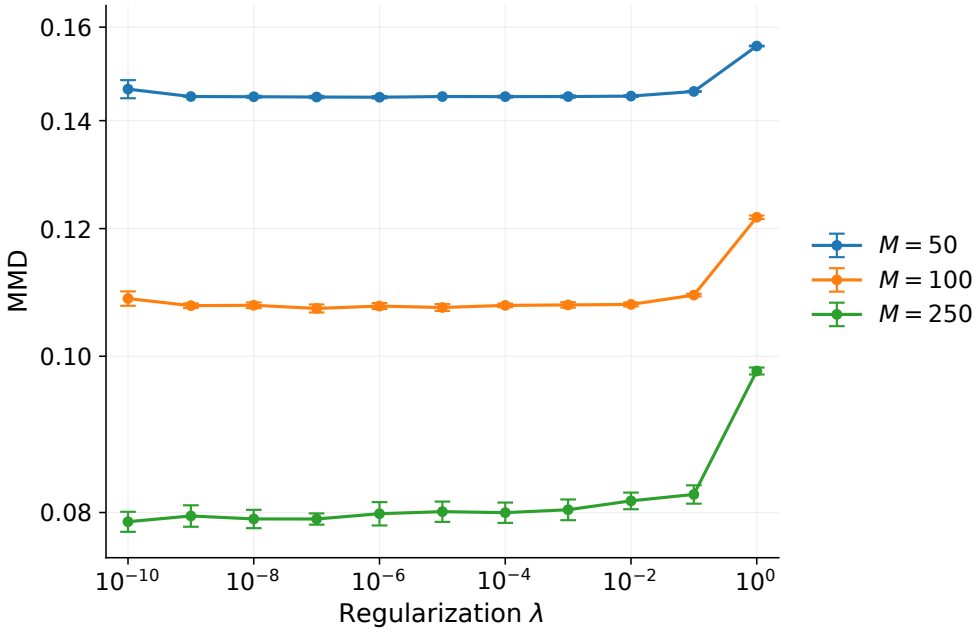


Figure 5: The sensitivity of the MMD to the level of regularization when MSIP-Fredholm is used.

**Empirical runtime scaling** We benchmark MSIP-Fredholm on a mixture of five two-dimensional Gaussians, with $M = 10 \cdot 2^j$ for $j = 0, \ldots, 8$. For each $M$, we report the minimum runtime over ten trials of 1000 iterations on an NVIDIA RTX 4090. The dashed curve shows the empirical fit $t_{\text{fit}}(M) = 0.7 + \alpha M^\beta$, $\alpha = 6.8 \times 10^{-5}$, $\beta = 1.768$, where runtime is measured in seconds. The offset is fixed at 0.7, and the remaining parameters are obtained by linear regression of $\log(t - 0.7)$

against $\log M$. This fit characterizes the observed scaling for this implementation and hardware over the tested range.

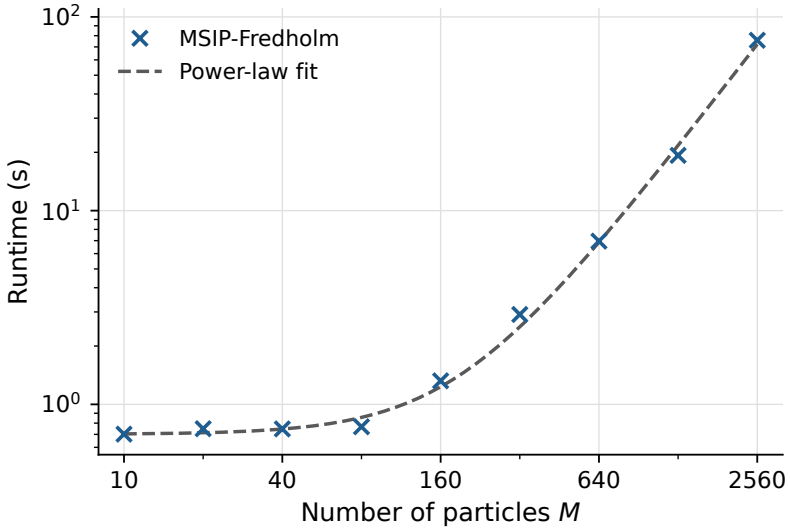


Figure 6: Runtime for 1000 iterations when MSIP-Fredholm is used.

# B Proofs

## B.1 Proof of Theorem 3.1

First, let $C_\pi := \int_{\mathcal{X}} \int_{\mathcal{X}} \kappa(x, x') \, \mathrm{d}\pi(x) \mathrm{d}\pi(x')$. Then, observe that for $Y \in \mathcal{X}^M$ we have

$$F_{M,\lambda}(Y) = \frac{1}{2}\Big( C_\pi - 2\langle \hat{\boldsymbol{w}}(Y), \boldsymbol{v}_0(Y) \rangle + \langle \hat{\boldsymbol{w}}(Y), \boldsymbol{K}_\lambda(Y) \hat{\boldsymbol{w}}(Y) \rangle \Big), \tag{42}$$

where $\hat{\boldsymbol{w}}(Y)$ is defined by (14) (omitting the $\lambda$ subscript for concision), and $v_0$ is given by (2). Using (14), we see that

$$\langle \hat{\boldsymbol{w}}(Y), \boldsymbol{v}_0(Y) \rangle = \langle \boldsymbol{v}_0(Y), \boldsymbol{K}_\lambda(Y)^{-1} \boldsymbol{v}_0(Y) \rangle = \langle \hat{\boldsymbol{w}}(Y), \boldsymbol{K}_\lambda(Y) \hat{\boldsymbol{w}}(Y) \rangle. \tag{43}$$

In particular, the expression of $F_{M,\lambda}(Y)$ simplifies further to

$$F_{M,\lambda}(Y) = \frac{1}{2}\Big( C_\pi - \langle \hat{\boldsymbol{w}}(Y), \boldsymbol{v}_0(Y) \rangle \Big) = \frac{1}{2}\Big( C_\pi - \langle \hat{\boldsymbol{w}}(Y), \boldsymbol{K}_\lambda(Y) \hat{\boldsymbol{w}}(Y) \rangle \Big).$$

In the following, we proceed to the calculation of the gradient of $F_M$. First, for a configuration $Y \in \mathcal{X}^M$, we define $y_{ij}$ as the scalar in the $j$th dimension of node $y_i$. Now, we seek the gradient $\nabla F_M \in \mathbb{R}^{M \times d}$. In particular, for $i \in [M]$ and $j \in [d]$, we use $\nabla_{ij} F_M(Y)$ to denote the $(i, j)$ entry of $\nabla F_M(Y)$ corresponding to input $y_{ij}$ evaluated at $Y \in \mathbb{R}^{M \times d}$. Moreover, we define $\nabla_{ij} \boldsymbol{K}_\lambda(Y) \in \mathbb{R}^{M \times M}$ to be the differentiation of each element of matrix $\boldsymbol{K}_\lambda(Y)(Y)$ with respect to the $j$th coordinate of vector $y_i$. Similarly, for kernel mean embedding $\boldsymbol{v}_0 : \mathcal{X}^M \to \mathbb{R}^M$ we denote by $\nabla_{ij} \boldsymbol{v}_0$ the vector obtained by differentiating each element of $\boldsymbol{v}_0$ with respect to the $j$th coordinate of vector $y_i$. In other words, we have

$$\nabla_{ij} F_{M,\lambda} = \frac{\partial}{\partial y_{ij}} F_{M,\lambda}(Y), \quad \nabla_{ij} \boldsymbol{K}_\lambda(Y) = \frac{\partial}{\partial y_{ij}} \boldsymbol{K}_\lambda(Y), \quad \nabla_{ij} \boldsymbol{v}(Y) = \frac{\partial}{\partial y_{ij}} \boldsymbol{v}(Y) \tag{44}$$

Using (43) and matrix calculus, we have

$$2\nabla_{ij} F_{M,\lambda}(Y) = -2\langle \hat{\boldsymbol{w}}(Y), \, \boldsymbol{K}_\lambda(Y) \nabla_{ij} \hat{\boldsymbol{w}}(Y) \rangle - \langle \hat{\boldsymbol{w}}(Y), \, \nabla_{ij} \boldsymbol{K}_\lambda(Y) \, \hat{\boldsymbol{w}}(Y) \rangle. \tag{45}$$

Since $\hat{\boldsymbol{w}}(Y) = \boldsymbol{K}_\lambda(Y)^{-1} \boldsymbol{v}_0(Y)$, we get

$$\nabla_{ij} \hat{\boldsymbol{w}}(Y) = \boldsymbol{K}_\lambda(Y)^{-1} \nabla_{ij} \boldsymbol{v}_0(Y) - \boldsymbol{K}_\lambda(Y)^{-1} \, \nabla_{ij} \boldsymbol{K}_\lambda(Y) \, \boldsymbol{K}_\lambda(Y)^{-1} \boldsymbol{v}_0(Y),$$

thus

$$\boldsymbol{K}_\lambda(Y)\nabla_{ij}\hat{\boldsymbol{w}}(Y) = \nabla_{ij}\boldsymbol{v}_0(Y) - \nabla_{ij}\boldsymbol{K}_\lambda(Y)\,\hat{\boldsymbol{w}}(Y),$$

so that

$$\langle \hat{\boldsymbol{w}}(Y),\, \boldsymbol{K}_\lambda(Y)\nabla_{ij}\hat{\boldsymbol{w}}(Y)\rangle = \langle \hat{\boldsymbol{w}}(Y),\, \nabla_{ij}\boldsymbol{v}_0(Y)\rangle - \langle \hat{\boldsymbol{w}}(Y),\, \nabla_{ij}\boldsymbol{K}_\lambda(Y)\hat{\boldsymbol{w}}(Y)\rangle. \tag{46}$$

Therefore, we have

$$\begin{aligned} 2\nabla_{ij}F_{M,\lambda}(Y) &= -2\Big(\langle \hat{\boldsymbol{w}}(Y),\, \nabla_{ij}\boldsymbol{v}_0(Y)\rangle - \langle \hat{\boldsymbol{w}}(Y),\, \nabla_{ij}\boldsymbol{K}_\lambda(Y)\,\hat{\boldsymbol{w}}(Y)\rangle\Big) \\ &\quad - \langle \hat{\boldsymbol{w}}(Y),\, \nabla_{ij}\boldsymbol{K}_\lambda(Y)\,\hat{\boldsymbol{w}}(Y)\rangle \\ &= -2\langle \hat{\boldsymbol{w}}(Y),\, \nabla_{ij}\boldsymbol{v}_0(Y)\rangle + \langle \hat{\boldsymbol{w}}(Y),\, \nabla_{ij}\boldsymbol{K}_\lambda(Y)\,\hat{\boldsymbol{w}}(Y)\rangle. \end{aligned} \tag{47}$$

Now recall that

$$[\nabla_{ij}\boldsymbol{K}_\lambda(Y)]_{m_1,m_2} = \nabla_{ij}\kappa(y_{m_1}, y_{m_2}),$$

for all $m_1,\, m_2 \in [M]$; the regularization parameter $\lambda$ disappears here as it is not a function of $Y$. Therefore, for the squared-exponential $\kappa$ with $\bar{\kappa}(x,y) := \sigma^{-2}\kappa(x,y)$,

$$\nabla_{ij}\kappa(y_{m_1}, y_{m_2}) = \delta_{i,m_1}(y_{m_2 j} - y_{ij})\bar{\kappa}(y_i, y_{m_2}) + \delta_{i,m_2}(y_{m_1 j} - y_{ij})\bar{\kappa}(y_{m_1}, y_i) \tag{48}$$

$$= \sigma^{-2}\Big(\delta_{i,m_1}(y_{m_2 j} - y_{ij})\kappa(y_i, y_{m_2}) + \delta_{i,m_2}(y_{m_1 j} - y_{ij})\kappa(y_{m_1}, y_i)\Big) \tag{49}$$

First, we denote $[\boldsymbol{a}\otimes\boldsymbol{b}]_{ij} = a_i b_j$ as the outer product between vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ with possibly different dimensions. We also denote $\boldsymbol{a}\odot\boldsymbol{b}$ as the elementwise (i.e., Hadamard) product between $\boldsymbol{a}$ and $\boldsymbol{b}$ with identical dimensions. Then, we have

$$\nabla_{ij}\boldsymbol{K}_\lambda(Y) = \boldsymbol{b}_{ij}\otimes\boldsymbol{e}_i + \boldsymbol{e}_i\otimes\boldsymbol{b}_{ij},$$

where $\boldsymbol{e}_i$ is the $i$th elementary vector, i.e., $[\boldsymbol{e}_i]_m = \delta_{im}$, and we use the expression of the squared exponential kernel to get definition

$$\boldsymbol{b}_{ij} := \sigma^{-2}(Y_j - y_{ij}\mathbf{1})\odot\boldsymbol{K}_i = \sigma^{-2}\Big(Y_j\odot\boldsymbol{K}_i - y_{ij}\boldsymbol{K}_i\Big),$$

with $Y_j\in\mathbb{R}^M$ is the vector containing the $j$-th entry of each $y_i$, identical to the $j$th column of the matrix $Y$, and $\boldsymbol{K}_i = \boldsymbol{K}(Y)\boldsymbol{e}_i$. Then,

$$\begin{aligned} \langle \hat{\boldsymbol{w}}(Y),\, \nabla_{ij}\boldsymbol{K}_\lambda(Y)\,\hat{\boldsymbol{w}}(Y)\rangle &= \langle \hat{\boldsymbol{w}}(Y),\, (\boldsymbol{b}_{ij}\otimes\boldsymbol{e}_i + \boldsymbol{e}_i\otimes\boldsymbol{b}_{ij})\hat{\boldsymbol{w}}(Y)\rangle \\ &= 2\hat{w}_i(Y)\langle \hat{\boldsymbol{w}}(Y),\, \boldsymbol{b}_{ij}\rangle \\ &= 2\sigma^{-2}\hat{w}_i(Y)\Big(\langle \hat{\boldsymbol{w}}(Y),\, Y_j\odot\boldsymbol{K}_i - y_{ij}\boldsymbol{K}_i\rangle\Big) \\ &= 2\sigma^{-2}\hat{w}_i(Y)\Big(\langle \hat{\boldsymbol{w}}(Y),\, Y_j\odot\boldsymbol{K}_i\rangle - y_{ij}\langle \hat{\boldsymbol{w}}(Y),\, \boldsymbol{K}_i\rangle\Big) \end{aligned} \tag{50}$$

For $m\in[M]$, we recall that $\nabla\kappa$ is bounded on $\mathcal{X}\times\mathcal{X}$ to prove

$$\begin{aligned} \nabla_{ij}\boldsymbol{v}_0(Y) &= (\nabla_{ij}v_0(y_i))\,\boldsymbol{e}_i \\ \nabla_{ij}v_0(y_i) &= \int_{\mathcal{X}}\nabla_{ij}\kappa(x,y_i)\,\mathrm{d}\pi(x) \\ &= \int_{\mathcal{X}}(x_j - y_{ij})\bar{\kappa}(x,y_i)\,\mathrm{d}\pi(x) \\ &= \sigma^{-2}\Big(\int_{\mathcal{X}}x_j\kappa(x,y_i)\,\mathrm{d}\pi(x) - y_{ij}v_0(y_i)\Big), \end{aligned}$$

where $x_j\in\mathbb{R}$ is the $j$th entry of integration variable $x\in\mathbb{R}^d$, and we recall that $v_0(y) = \int_{\mathcal{X}}\kappa(x,y)\,\mathrm{d}\pi(x)$. This equivalence is precisely due to the fact that $\kappa\propto\bar{\kappa}$. For derivative kernel $\bar{\kappa}$ as discussed in [11], this identity would not hold. We then have

$$\nabla_{ij}\boldsymbol{v}_0(Y) = \sigma^{-2}\left([v_1(y_i)]_j - y_{ij}v_0(y_i)\right)\boldsymbol{e}_i.$$

Thus,

$$\langle \hat{\boldsymbol{w}}(Y),\, \nabla_{ij}\boldsymbol{v}_0(Y)\rangle = \sigma^{-2}\hat{w}_i(Y)\left([v_1(y_i)]_j - y_{ij}v_0(y_i)\right). \tag{51}$$

Combining (47), (50) and (51), we obtain

$$\begin{aligned} 2\nabla_{ij} F_{M,\lambda}(Y) &= 2\sigma^{-2}\hat{w}_i(Y)\left(-[v_1(y_i)]_j + y_{ij}v_0(y_i) + \langle \hat{\boldsymbol{w}}(Y),\, Y_j \odot \boldsymbol{K}_i\rangle - y_{ij}\langle \hat{\boldsymbol{w}}(Y),\, \boldsymbol{K}_i\rangle\right) \\ &= 2\sigma^{-2}\hat{w}_i(Y)\left(-[v_1(y_i)]_j + y_{ij}v_0(y_i) - y_{ij}\langle \hat{\boldsymbol{w}}(Y),\, \boldsymbol{K}_i\rangle + \langle \hat{\boldsymbol{w}}(Y),\, Y_j \odot \boldsymbol{K}_i\rangle\right) \end{aligned} \tag{52}$$

Now,

$$\begin{aligned} y_{ij}v_0(y_i) - y_{ij}\langle \hat{\boldsymbol{w}}(Y),\, \boldsymbol{K}_i\rangle &= y_{ij}\left(-\sum_{m=1}^{M}\hat{w}_m(Y)\kappa(y_i,y_m) + \int_{\mathcal{X}}\kappa(x,y_i)\,\mathrm{d}\pi(x)\right) \\ &= \lambda\hat{w}_i(Y)y_{ij} \qquad (53) \\ &= \left[\lambda \boldsymbol{W}(Y)Y\right]_{i,j}. \qquad (54) \end{aligned}$$

The identity used in (53) is precisely a rearrangement of the fact that $\boldsymbol{K}(Y)\hat{\boldsymbol{w}} + \lambda\hat{\boldsymbol{w}} = \boldsymbol{v}_0(Y)$. Using the symmetry of $\kappa$,

$$\begin{aligned} \langle \hat{\boldsymbol{w}}(Y),\, Y_j \odot \boldsymbol{K}_i\rangle &= \sum_{m=1}^{M}\hat{w}_m(Y)y_{mj}\kappa(y_m,y_i) \\ &= \sum_{m=1}^{M}\kappa(y_i,y_m)\hat{w}_m(Y)y_{mj} \\ &= \langle \boldsymbol{e}_i,\, \boldsymbol{K}(Y)\boldsymbol{W}(Y)Y_j\rangle \\ &= \left[\boldsymbol{K}(Y)\boldsymbol{W}(Y)Y\right]_{i,j} \end{aligned} \tag{55}$$

where $\boldsymbol{W}(Y) \in \mathbb{R}^{M\times M}$ satisfies $[\boldsymbol{W}(Y)]_{i,m} = \hat{w}_i(Y)\delta_{i,m}$. Combining (55) and the fact that $[v_1(y_i)]_j$ is the $(i,j)$-th entry of the matrix $\boldsymbol{v}_1(Y)$, we conclude that

$$\left(-[v_1(y_i)]_j + y_{ij}v_0(y_i) - y_{ij}\langle \hat{\boldsymbol{w}}(Y),\, \boldsymbol{K}_i\rangle + \langle \hat{\boldsymbol{w}}(Y),\, Y_j \odot \boldsymbol{K}_i\rangle\right)$$

is the $(i,j)$-entry of the matrix $-\boldsymbol{v}_1(Y) + \boldsymbol{K}(Y)\boldsymbol{W}(Y)Y + \lambda\boldsymbol{W}(Y)Y = -\boldsymbol{v}_1(Y) + \boldsymbol{K}_\lambda(Y)\boldsymbol{W}(Y)Y$. Thus, (52) is the $(i,j)$-entry of the matrix

$$2\sigma^{-2}\boldsymbol{W}(Y)\left(\boldsymbol{K}_\lambda(Y)\boldsymbol{W}(Y)Y - \boldsymbol{v}_1(Y)\right). \tag{56}$$

In other words,

$$\nabla F_M(Y) = \sigma^{-2}\boldsymbol{W}(Y)\left(\boldsymbol{K}_\lambda(Y)\boldsymbol{W}(Y)Y - \boldsymbol{v}_1(Y)\right). \tag{57}$$

When the weights $\boldsymbol{W}(Y)$ are nonzero, we thus have

$$\nabla F_M(Y) = \sigma^{-2}\boldsymbol{W}(Y)\boldsymbol{K}_\lambda(Y)\boldsymbol{W}(Y)\left(Y - \boldsymbol{W}(Y)^{-1}\boldsymbol{K}_\lambda(Y)^{-1}\boldsymbol{v}_1(Y)\right). \tag{58}$$

We remark that $\boldsymbol{K}_\lambda(Y)$ is invertible by construction for any $\lambda > 0$. Finally, observe that the function $\boldsymbol{\Psi}_{\lambda,\mathrm{MSIP}}$ defined by (13) satisfies

$$\boldsymbol{\Psi}_{\mathrm{MSIP}}(Y) = \boldsymbol{W}(Y)^{-1}\boldsymbol{K}_\lambda(Y)^{-1}\boldsymbol{v}_1(Y). \tag{59}$$

### B.2 Proof of Proposition 3.2

First, fix $C > 0$ and define scaled density $\tilde{\pi} = C\pi$ for convenience. Then, given $Y \in \mathcal{X}^M$, we have

$$\Psi_{\mathrm{MSIP},\lambda}(Y;\pi) = \boldsymbol{W}_\lambda^*(Y)^{-1}\boldsymbol{K}_\lambda(Y)^{-1}\boldsymbol{V}_0(Y)r(Y), \tag{60}$$

where $\boldsymbol{V}_0(Y) = \mathrm{diag}(\boldsymbol{v}_0(Y))$, and $r : \mathcal{X}^M \to \mathcal{X}^M$ has row $i$ defined by $v_1(y_i)/v_0(y_i)$. Fix $Y \in \mathcal{X}^M$, then define the functions $v_0^C$ and $v_1^C$ for $\tilde{\pi}$, corresponding to $v_0$ and $v_1$, respectively. Remarkably, we observe

$$v_0^C(y) = C\int \kappa(x,y)\,\mathrm{d}\pi(x) = Cv_0(y), \tag{61}$$

$$v_1^C(y) = C\int x\kappa(x,y)\,\mathrm{d}\pi(x) = Cv_1(y). \tag{62}$$

This identity also holds for any of our estimators $\hat{v}_0$ and $\hat{v}_1$. Thus,

$$\frac{v_1^C(y)}{v_0^C(y)} = \frac{v_1(y)}{v_0(y)}. \tag{63}$$

Stated plainly, the vector of ratios $r(Y)$ is invariant to the value of $C$. Moreover, the weights $\hat{\boldsymbol{w}}_\lambda^C$ for $\tilde{\pi}$ satisfy

$$\boldsymbol{w}_\lambda^{*C}(Y) := \boldsymbol{K}_\lambda(Y)^{-1}\boldsymbol{v}_0^C(Y) = C\boldsymbol{w}_\lambda^*(Y), \tag{64}$$

so that the product $\boldsymbol{W}_\lambda^*(Y)^{-1}\boldsymbol{K}_\lambda(Y)^{-1}\boldsymbol{V}_0(Y)$ is invariant to the value of $C$. Therefore, $\Psi_{\mathrm{MSIP},\lambda}(Y; C\cdot\pi)$ is invariant to the value of $C$.

### B.3 One-point MSIP map as a GMM approximation

**Proposition B.1.** *Suppose the target density $\pi$ is in RKHS $\mathcal{H}$ with squared-exponential kernel. Evaluate $\widehat{\Psi}_{\mathrm{MSIP},\lambda}$ using one-point estimators $\hat{v}_0^1$ and $\hat{v}_1^{S1}$ in* (30)*. If there exists a measure $\mu$ with*

$$\pi(x) = \int \kappa(x,y)\,\mathrm{d}\mu(y), \tag{65}$$

*then any fixed point of $\widehat{\Psi}_{\mathrm{MSIP},\lambda}$ is a critical point of $F_{M,\lambda}^\mu$. Further, we have the identity*

$$F_{M,\lambda}^\mu(Y) \equiv \min_{\boldsymbol{w}\in\mathbb{R}^M} \left\| \pi - \sum_{i=1}^M w_i \mathcal{N}(\cdot; y_i,\ \sigma^2 I) \right\|_{\mathcal{H}}^2 + \frac{\lambda}{2}\|\boldsymbol{w}\|^2.$$

*Proof.* We remark that the MSIP map $\Psi_{\mathrm{MSIP},\lambda}$ defined using $\boldsymbol{v}_0$ and $\boldsymbol{v}_1$ for the measure $\mu$ precisely aligns with the practical calculation of $\hat{v}_0^1$ and $\hat{v}_1^{S1}$. Therefore, we apply Theorem 3.1 to arrive at the first half of the result.

Now, consider a configuration of points and weights $Y, \boldsymbol{w}$. Let $f_i(x) = \mathcal{N}(x; y_i, \sigma^2 I)$ and note that

$$\pi(x) - \sum_{i=1}^M w_i f_i(x) = \left\langle \kappa(x,\cdot), \mu - \sum_{i=1}^M w_i \delta_{y_i} \right\rangle = \Phi_{\mathcal{H}}\left[\mu - \sum_{i=1}^M w_i \delta_{y_i}\right](x),$$

where $\Phi_{\mathcal{H}}$ is the feature map of RKHS $\mathcal{H}$ [64]Therefore, we see that

$$\left\| \pi - \sum_{i=1}^M w_i \mathcal{N}(\cdot; y_i,\ \sigma^2 I) \right\|_{\mathcal{H}}^2 = \left\| \Phi_{\mathcal{H}}\left[\mu - \sum_{i=1}^M w_i \delta_{y_i}\right] \right\|_{\mathcal{H}}^2 = \mathrm{MMD}^2\left(\mu, \sum_{i=1}^M w_i \delta_{y_i}\right).$$

The result thus follows from the definition of $F_{M,\lambda}^\mu$. □

### B.4 2D visualisations

In the following figures, we include a slew of visualizations of final configurations of our proposed methods compared to others. The algorithms include the ones benchmarked against in the main paper, as well as multi-start gradient ascent (GA). One can see that GA has significant problems capturing anisotropy within the modes. Further, we distinguish between SVGD with a fixed bandwidth (here, denoted simply SVGD) versus SVGD with an adaptive bandwidth (a-SVGD) through the median heuristic, which is used in the main experiments.

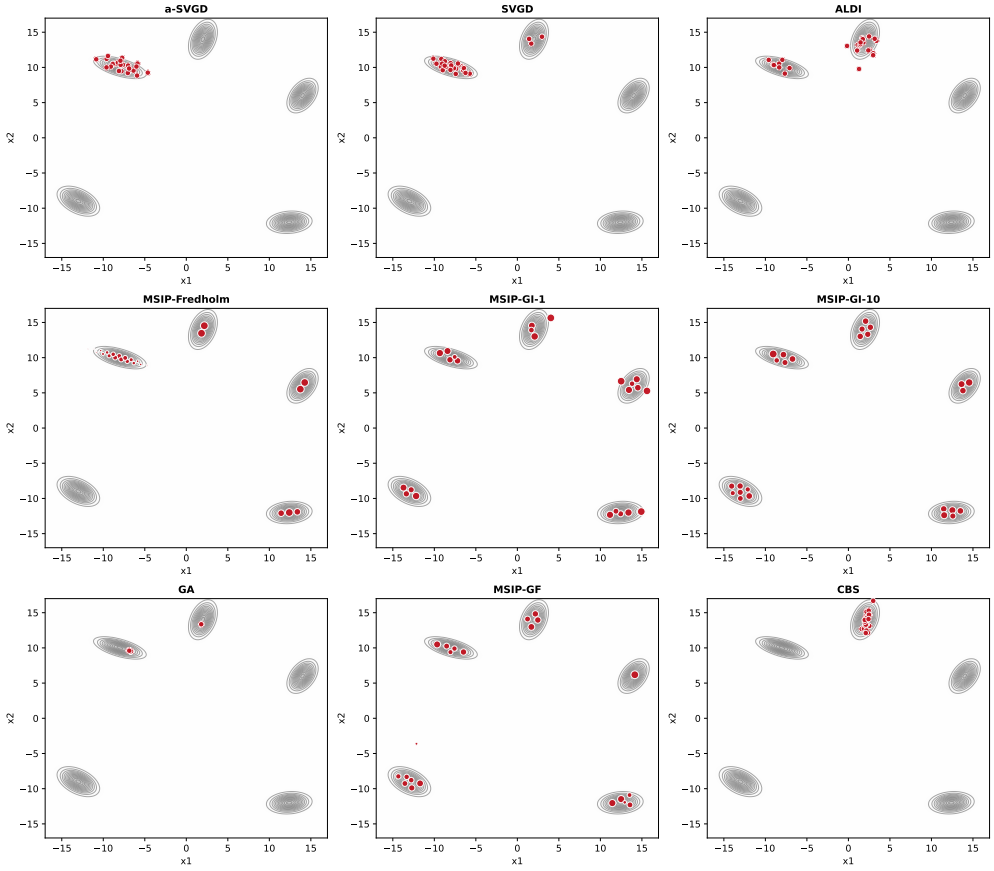


Figure 7: GMM 2D

### B.5 Additional numerics

Table 4: Error of different algorithms for synthetic distributions using KSD with an inverse multi-quadric kernel for *unweighted particles*. We report mean and standard deviation over many trials; the smallest row-wise mean is bolded. Empty cells correspond to an algorithm not converging.

| | | Gradient-informed algorithms | | | | | Gradient-free algorithms | |
|---|---|---|---|---|---|---|---|---|
| | Dim. | **MSIPF** | **MSIPGI-1** | **MSIPGI-10** | SVGD | ALDI | **MSIPGF** | CBS |
| GMM2 | 2 | **0.794 (0.025)** | 1.249 (0.366) | 1.658 (2.049) | 1.223 (0.039) | 0.930 (0.094) | 25.54 (22.15) | 1.103 (0.090) |
| GMM5 | 5 | 1.378 (0.029) | 3.014 (3.674) | **1.310 (0.001)** | 1.499 (0.004) | 1.516 (0.054) | 19.95 (40.99) | 3.293 (0.336) |
| GMM10 | 10 | 2.228 (9.064e-04) | **1.965 (0.010)** | 1.972 (0.022) | 2.172 (0.002) | 2.185 (0.039) | 8.774 (8.066) | 3.899 (0.369) |
| GMM20 | 20 | 3.395 (8.309e-06) | 3.018 (0.031) | **2.912 (0.044)** | 3.060 (0.003) | 3.355 (0.171) | 2.938 (0.018) | 5.061 (0.345) |
| Joker | 2 | 8.447 (0.028) | 8.401 (0.016) | **8.394 (0.026)** | 11.11 (1.579) | 9.286 (0.185) | 11.98 (7.794) | 10.02 (1.475) |
| Funnel10 | 10 | 20.24 (0.041) | 20.24 (0.039) | 20.24 (0.040) | 24.06 (1.593) | 20.34 (0.167) | **20.20 (0.026)** | 21.86 (1.396) |
| Funnel20 | 20 | 28.93 (0.140) | 28.92 (0.163) | 28.94 (0.165) | 67.36 (22.99) | **28.66 (0.122)** | 28.74 (0.101) | 34.05 (3.475) |
| Funnel 2D | 2 | 8.628 (0.021) | 8.606 (0.023) | **8.600 (0.020)** | 9.073 (0.145) | 9.008 (0.091) | 8.755 (0.019) | 95.40 (176.8) |
| Funnel50 | 50 | 47.78 (2.888) | 46.70 (0.741) | 46.82 (0.785) | 242.5 (142.7) | — (—) | **45.54 (0.132)** | 58.30 (10.52) |
| Funnel5 | 5 | 14.25 (0.021) | 14.25 (0.021) | 14.25 (0.020) | 18.64 (2.574) | 14.43 (0.210) | **14.24 (0.017)** | 14.98 (0.615) |
| Himmelblau | 2 | 7.792 (0.239) | 9.050 (1.389) | **7.389 (0.080)** | 14.24 (0.311) | 14.33 (2.558) | 8.971 (2.273) | — (—) |

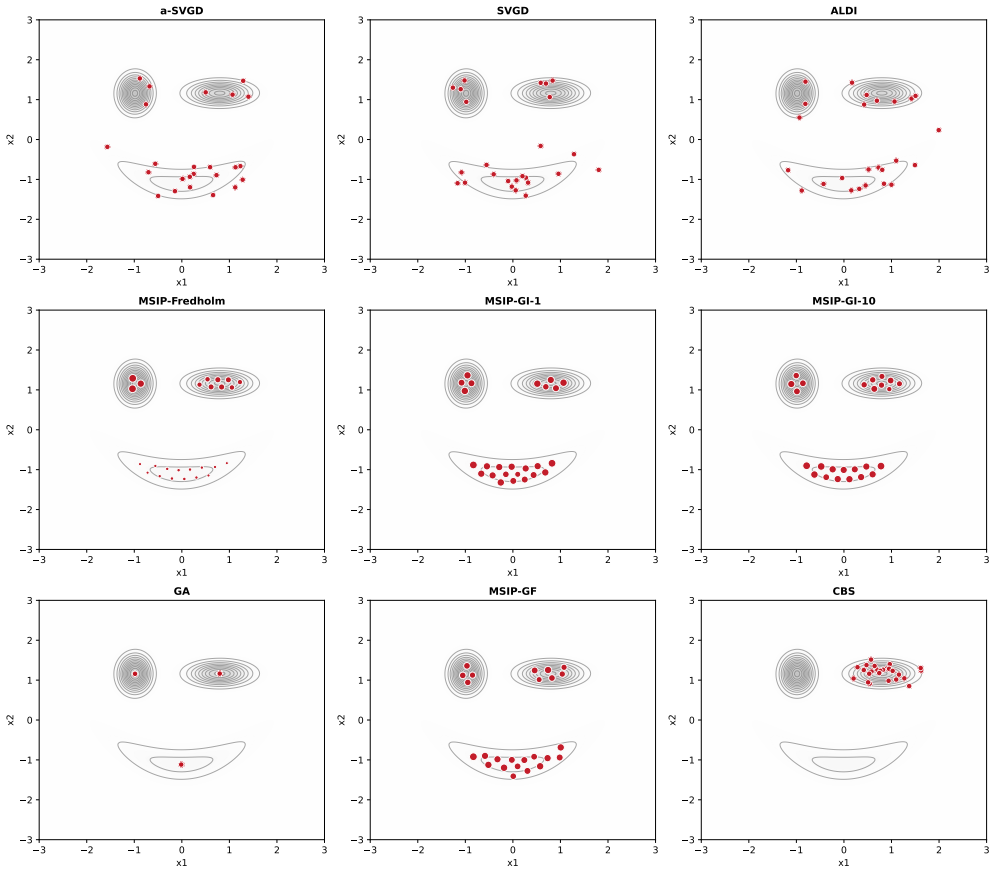


Figure 8: Joker

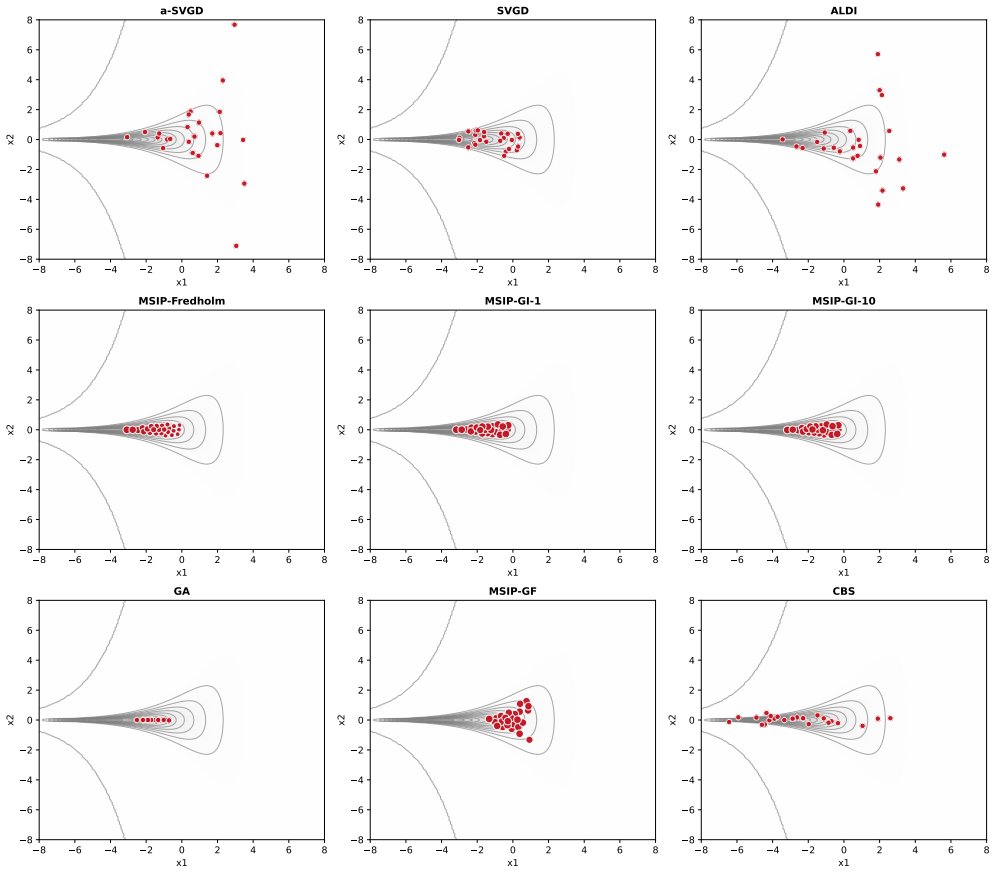


Figure 9: Funnel2D

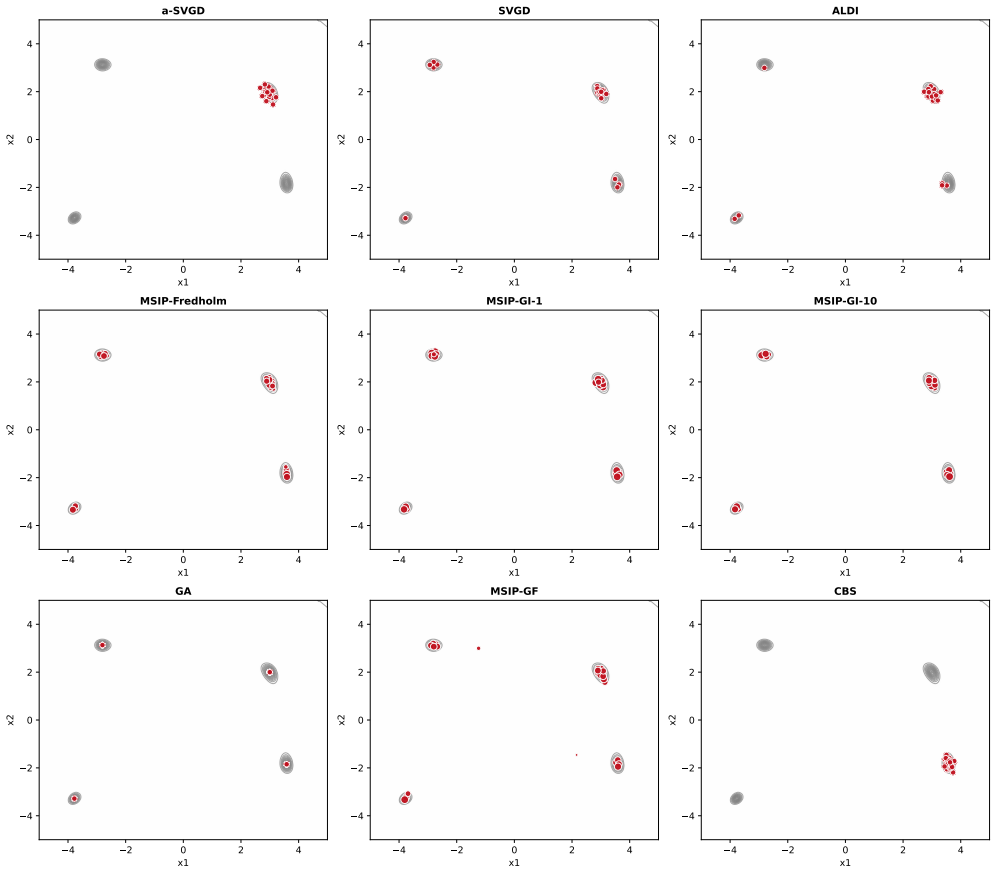


Figure 10: Himmelblau

## NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We demonstrate the proposed contributions using a combination of proofs and numerical examples.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We show that, depending on the metric you care about, other methods can outperform our contributions.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: We show these proofs in Section B.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: See Table 3 and the containing section.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We have provided the relevant portion of anonymized code in supplementary materials.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: See Section A.3.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We run several trials for each result and show some uncertainty induced by these trials in our results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See Section A.3.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have reviewed these guidelines and affirm that we conform to them.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: There is no clear line between our contributions at the scale we show and societal impacts—our methods do not do purport to do anything that other typical Bayesian inference methods do not attempt to.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Our paper poses no such risk.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite the appropriate sources of the code and data where necessary.

    Guidelines:

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

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We use a suite of examples to show the code performance and coherence.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs.

Guidelines:

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