Weighted Quantization Using MMD: From Mean Field to Mean Shift via Gradient Flows

Ayoub Belhadji **Daniel Sharp** **Youssef Marzouk**
Center for Computational Science and Engineering
Massachusetts Institute of Technology

Abstract

Approximating a probability distribution using a set of particles is a fundamental problem in machine learning and statistics, with applications including clustering and quantization. Formally, we seek a weighted mixture of Dirac measures that best approximates the target distribution. While much existing work relies on the Wasserstein distance to quantify approximation errors, maximum mean discrepancy (MMD) has received comparatively less attention, especially when allowing for variable particle weights. We argue that a *Wasserstein–Fisher–Rao* gradient flow is well-suited for designing quantizations optimal under MMD. We show that a system of interacting particles satisfying a set of ODEs discretizes this flow. We further derive a new fixed-point algorithm called *mean shift interacting particles* (MSIP). We show that MSIP extends the classical mean shift algorithm, widely used for identifying modes in kernel density estimators. Moreover, we show that MSIP can be interpreted as preconditioned gradient descent and that it acts as a relaxation of Lloyd's algorithm for clustering. Our unification of gradient flows, mean shift, and MMD-optimal quantization yields algorithms more robust than state-of-the-art methods, as demonstrated via high-dimensional and multimodal numerical experiments.

1 INTRODUCTION

Numerous problems in statistics and machine learning involve approximating a probability measure using a small set of points, a task usually referred to as quantization (Graf and Luschgy, 2000). Examples include clustering (Lloyd, 1982), numerical integration (Robert et al., 1999; Bach, 2017), and teacher-student training of neural networks (Chizat et al., 2019; Arbel et al., 2019). Formally, the quantization problem consists of seeking an approximation to a probability measure π using a mixture of Dirac measures $\sum_{i=1}^{M} w_i \delta_{y_i}$, where the y_i are usually called nodes or centroids and the w_i are weights. In this work, we focus on the problem

$$\min_{\mu \in \mathrm{Mix}_M(\mathcal{X})} \mathrm{MMD}\left(\pi, \mu\right), \tag{1}$$

where $\mathrm{Mix}_M(\mathcal{X}) := \left\{ \sum_{i=1}^{M} w_i \delta_{y_i} \,\middle|\, w_i \in \mathbb{R},\, y_i \in \mathcal{X} \right\}$ and MMD is the maximum mean discrepancy defined as follows: given a measurable space $\mathcal{X}$ and a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, the MMD between measures μ and ν is defined as

$$\mathrm{MMD}(\mu, \nu) := \sup_{\|f\|_{\mathcal{H}} \leq 1} \left| \int_{\mathcal{X}} f(x) \mathrm{d}\mu(x) - \int_{\mathcal{X}} f(x) \mathrm{d}\nu(x) \right|.$$

Initially introduced for statistical analysis and hypothesis testing (Gretton et al., 2007, 2012), the MMD is applied in diverse fields including generative modeling (Dziugaite et al., 2015; Li et al., 2017) and optimal transport (Chewi et al., 2024; Peyré and Cuturi, 2019).

Lloyd's algorithm (Lloyd, 1982) is a very popular quantization method, but seeks to minimize a Wasserstein distance instead of solving problem (1). Moreover, while many classical formulations of quantization constrain the weights w_i to lie on the simplex, the minimization problem (1) relaxes this requirement. Our approach leverages the natural connection between this minimization problem and gradient flows, a topic that has seen substantial theoretical and algorithmic progress in recent years. While MMD minimization using gradient flows has already been studied (Arbel et al., 2019; Gladin et al., 2024), most existing work considers the number of atoms tending to infinity, i.e., the mean-field limit. This conflicts with the fundamental motivation of quantization, and results in poor

---

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

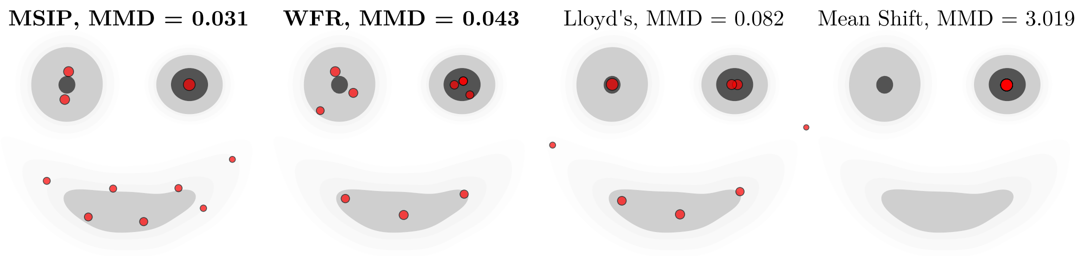


Figure 1: Comparison of quantization algorithms on a joker distribution, $M = 10$. All algorithms initialized identically on the top-right mode. Each marker's size denotes the relative particle weight.

performance when the number of atoms is small. Our inspiration comes from considering problem (1) at the opposite extreme, i.e., a single atom ($M = 1$). This corresponds to seeking the mode of a distribution using the *mean shift* algorithm (Fukunaga and Hostetler, 1975). Mean shift is defined exclusively for this single particle case. Remarkably, the mean shift algorithm is a fixed-point iteration that we can also interpret as a preconditioned gradient ascent of the kernel density estimator (KDE), where the preconditioner helps the particle escape from low density regions.

Building on these observations, we *unify* views of quantization, gradient flows, and mode-seeking for MMD minimization using a *finite* set of weighted particles. Concretely, our **main contributions** are as follows:

- We tackle the minimization problem (1) by first adopting the perspective of gradient flows, where the MMD is a functional to be minimized by following the local direction of steepest descent in some chosen geometry. In particular, we propose using the Wasserstein–Fisher–Rao (WFR) geometry to allow for both transport and the creation and destruction of mass.
- We show that this approach admits efficient numerical implementation via an interacting particle system (IPS), described by ODEs for the time evolution of particle positions and weights. We characterize the infinite-time limit of this WFR-IPS and introduce an algebraic system, depending on kernelized moments of the target π, that yields stationary solutions.
- To directly find particles that satisfy this stationarity condition, we introduce a damped fixed-point algorithm termed *mean shift interacting particles* (MSIP), which we show to extend the classical mean shift algorithm. We further show that MSIP is a *preconditioned gradient descent* of

$$(y_1, \dots, y_M) \mapsto \min_{w_i \in \mathbb{R}} \frac{1}{2} \mathrm{MMD}^2 \left(\pi, \sum_{i=1}^{M} w_i \delta_{y_i} \right),$$

paralleling literature on Lloyd's algorithm (Du et al., 1999). In other words, MSIP minimizes MMD analogously to how Lloyd's algorithm minimizes Wasserstein distance.

- We demonstrate the computational *efficiency* and *robustness* of the proposed approaches for high-dimensional quantization problems, improving on current state-of-the-art methods. In particular, both MSIP and the WFR-IPS yield *near-optimal* quantizations under MMD, even with extremely adversarial initializations.

As a preview of our approach, Figure 1 compares the quantizations obtained using four algorithms: MSIP, WFR-IPS, Lloyd's algorithm, and classical mean shift. We see that MSIP and WFR-IPS identify high-density regions of the distribution while capturing the anisotropy in all three modes. Mean shift, however, completely collapses to a single mode. While Lloyd's algorithm also captures anisotropy, it tends to capture the support of the *entire* distribution instead of just high-probability regions. Finally, MSIP recognizes that the top-right mode is significantly concentrated and is adequately characterized with a single point; WFR-IPS and Lloyd instead over-represent variance within this mode. The quantizations' MMD values, shown at top, reflect these characteristics.

This article is structured as follows. In Section 2, we review the literature on numerical integration in an RKHS, quantization, the mean shift algorithm, and gradient flows. In Section 3 we present the main contributions of this work. In Section 4, we report numerical experiments that illustrate the proposed algorithms and compare to other schemes. Section 5 is a short concluding discussion.

Notation We denote by $\mathcal{X}$ a subset of $\mathbb{R}^d$, and we use $\mathcal{M}(\mathcal{X})$ and $\mathcal{P}(\mathcal{X})$ to refer to the set of measures and probability measures supported on $\mathcal{X}$, respectively. Moreover, given $M \in \mathbb{N}^*$, we denote by $\mathcal{X}^M$ the M-fold Cartesian product of $\mathcal{X}$.

We consider $Y \in \mathcal{X}^M$ as a configuration of M points $y_1, \dots, y_M \in \mathcal{X}$. Since $\mathcal{X} \subset \mathbb{R}^d$, we abuse notation and let Y refer to the $M \times d$ matrix whose rows are the vectors $y_1, \dots, y_M$. We also denote by Δ_{M-1} the simplex of dimension $M-1$ defined by $\{\boldsymbol{w} \in [0,1]^M \mid \sum_{i\in[M]} w_i = 1\}$. Consider $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ as the kernel associated to the RKHS $\mathcal{H}$. For a configuration Y, we denote by $\boldsymbol{K}(Y) := (\kappa(y_i, y_j))_{i,j\in[M]}$ the associated kernel matrix for $i, j \in [M]$. Additionally, for a given real-valued function $f : \mathcal{X} \to \mathbb{R}$, we denote by $\boldsymbol{f}(Y) := (f(y_i))_{i\in[M]}$ the vector of evaluations of f on the elements of Y. Finally, for a given vector-valued function $\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^d$, we denote by $\boldsymbol{g}(Y)$ the $M \times d$ matrix where each row is an evaluation of $\boldsymbol{g}$ on each y_i.

2 BACKGROUND AND RELATED WORK

The quantization problem (1) is crucial in the design of efficient numerical integration methods. Indeed, for a given probability measure $\pi \in \mathcal{P}(\mathcal{X})$ and a function $f : \mathcal{X} \to \mathbb{R}$, a quadrature rule seeks to approximate $\mathbb{E}_\pi[f] := \int_{\mathcal{X}} f(x)\,\mathrm{d}\pi(x)$ with $\sum_{i=1}^M w_i f(y_i)$, where the points $y_i \in \mathcal{X}$ are the quadrature nodes and the scalars w_i are the quadrature weights. When the function f belongs to the RKHS $\mathcal{H}$, the integration error is bounded as follows (Muandet et al., 2017):

$$\left|\mathbb{E}_\pi[f] - \sum_{i=1}^M w_i f(y_i)\right| \leq \|f\|_{\mathcal{H}} \mathrm{MMD}\Big(\pi, \sum_{i=1}^M w_i \delta_{y_i}\Big). \tag{2}$$

In other words, having an accurate quantization with respect to MMD leads to a highly accurate quadrature rule $\sum_{i=1}^M w_i \delta_{y_i}$ for functions living in the RKHS. Within this framework, the design of the quadrature rule remains independent of the function f, which is particularly advantageous in scenarios where evaluating f is expensive.

The *optimal kernel-based quadrature* is a popular and well-studied family of quadrature rules: for a given configuration of nodes Y, the weights $\hat{w}_1(Y), \dots, \hat{w}_M(Y)$ satisfy

$$\mathrm{MMD}\left(\pi, \sum_{i=1}^M \hat{w}_i(Y)\delta_{y_i}\right) \leq \mathrm{MMD}\left(\pi, \sum_{i=1}^M w_i \delta_{y_i}\right), \tag{3}$$

for any $w_1, \dots, w_M \in \mathbb{R}$. These optimal weights can be expressed as the entries of the vector

$$\hat{\boldsymbol{w}}(Y) = \boldsymbol{K}(Y)^{-1}\boldsymbol{v}_0(Y), \tag{4}$$

where $v_0 : \mathcal{X} \to \mathbb{R}$ is the *kernel mean embedding* of the probability measure π defined by

$$v_0(y) := \int_{\mathcal{X}} \kappa(x, y)\,\mathrm{d}\pi(x). \tag{5}$$

While the optimal weights $\hat{\boldsymbol{w}}(Y)$ have an analytical formula for fixed configuration Y, the function $Y \mapsto \mathrm{MMD}^2\left(\pi, \sum_{i=1}^M \hat{w}_i(Y)\delta_{y_i}\right)$ is daunting to minimize due to its non-convexity (Oettershagen, 2017). Recent work on optimal configurations often propose approaches tailored to specific standard probability measures (Karvonen and Särkkä, 2018, 2019; Ehler et al., 2019). Some other methods are designed to be universal, but their numerical implementation is generally limited by difficulties in high-dimensional domains. Examples include ridge leverage score sampling (Bach, 2017), determinantal point processes and volume sampling (Belhadji et al., 2019, 2020; Belhadji, 2021), Fekete points (Karvonen et al., 2021), randomly pivoted Cholesky (Epperly and Moreno, 2023), or greedy selection algorithms (De M., 2003; De M. et al., 2005; Santin and Haasdonk, 2017; Oettershagen, 2017; Huszár and Duvenaud, 2012; Briol et al., 2015; Lacoste-Julien et al., 2015; Teymur et al., 2021).

The function (5) is only practically tractable for some kernels and probability measures. One such case is when π is an empirical measure, where the quantization can be seen as a summarization of π; this is notably relevant for Bayesian inference (Owen, 2017; Riabiz et al., 2022). There are many approaches to this problem. Kernel thinning (Dwivedi and Mackey, 2022, 2024) is mainly concerned with uniformly weighted quantization, where one can only choose M if it satisfies $\lfloor N/2^r \rfloor$ for some positive integer r. Other methods such as kernel recombination (Hayakawa et al., 2022, 2023) or approximate ridge leverage score sampling (Chatalic et al., 2023) primarily focus on convergence rates in M, but are often outperformed in practice by simpler (e.g., greedy) methods when M is small.

2.1 Clustering

A standard method for quantization (Lloyd, 1982) is Lloyd's algorithm, which aims to solve the problem

$$\min_{Y\in\mathcal{X}^M} \min_{\boldsymbol{w}\in\Delta_{M-1}} \frac{1}{2} W_2^2\Big(\pi, \sum_{i=1}^M w_i \delta_{y_i}\Big), \tag{6}$$

where W_2 is the 2-Wasserstein distance (Peyré and Cuturi, 2019). The solution of (6) is naturally expressed in terms of *Voronoi tessellations* (Du et al., 1999). The Voronoi tesselation of a configuration $Y \in \mathcal{X}^M$ corresponds to the sets $\mathcal{V}_1(Y), \dots, \mathcal{V}_M(Y)$ defined by $\mathcal{V}_i(Y) := \left\{\, x \in \mathcal{X} \mid i = \arg\min_{m\in[M]} \|x - y_m\| \,\right\}$.

The algorithm then consists of taking the fixed-point iteration $Y^{(t+1)} = \boldsymbol{\Psi}_{\mathrm{Lloyd}}(Y^{(t)})$, where $\boldsymbol{\Psi}_{\mathrm{Lloyd}} : \mathcal{X}^M \to \mathcal{X}^M$ is the map defined by

$$\forall i \in [M],\ \ (\boldsymbol{\Psi}_{\mathrm{Lloyd}}(Y))_{i,:} := \frac{\int_{\mathcal{V}_i(Y)} x\mathrm{d}\pi(x)}{\int_{\mathcal{V}_i(Y)} \mathrm{d}\pi(x)}. \tag{7}$$

The fixed points of $\mathbf{\Psi}_{\text{Lloyd}}$ are critical points of the function G_M, defined as (Du et al., 1999, Proposition 6.2.)

$$G_M : Y \mapsto \min_{\boldsymbol{w}\in\Delta_{M-1}} \frac{1}{2} W_2^2\Big(\pi, \sum_{i\in[M]} w_i \delta_{y_i}\Big). \tag{8}$$

The authors prove that G_M is differentiable in Y with the gradient given by

$$\nabla G_M(Y) = \boldsymbol{D}(Y)(Y - \mathbf{\Psi}_{\text{Lloyd}}(Y)), \tag{9}$$

where $\boldsymbol{D}(Y) \in \mathbb{R}^{M\times M}$ is the diagonal matrix $(\boldsymbol{D}(Y))_{i,i} := \int_{\mathcal{V}_i(Y)} \mathrm{d}\pi(x)$. Hence, the fixed-point iteration is equivalent to preconditioned gradient descent, i.e.,

$$Y^{(t+1)} = Y^{(t)} - \boldsymbol{D}(Y^{(t)})^{-1} \nabla G_M(Y^{(t)}). \tag{10}$$

The theoretical analysis and nondegeneracy of Lloyd's algorithm has been the subject of extensive study (Kieffer, 1982; Wu, 1992; Du et al., 1999, 2006; Emelianenko et al., 2008; Pagès and Yu, 2016; Portales et al., 2025).

2.2 Mean shift

Mean shift is a widely used algorithm for tasks such as clustering (Fukunaga and Hostetler, 1975; Carreira-Perpinán, 2015) and image segmentation (Comaniciu and Meer, 2002). Mean shift locates the mode of a kernel density estimator (KDE)

$$\hat{f}_{\text{KDE}}(y) := \frac{1}{N}\sum_{\ell=1}^{N} \kappa(y, x_\ell) = v_0(y), \tag{11}$$

where v_0 is defined by (5), taking π to be the empirical measure associated to the data points $x_1, \ldots, x_N \in \mathcal{X}$. In this context, the kernel κ must satisfy the following.

Assumption 2.1. The kernel κ is $\mathcal{C}^1$ and there is a symmetric kernel $\bar{\kappa} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ satisfying

$$\forall x, y \in \mathcal{X}, \quad \nabla_2 \kappa(x, y) = (x - y)\bar{\kappa}(x, y). \tag{12}$$

Under Assumption 2.1, a critical point $y_\star$ of the KDE satisfies $\sum_{\ell=1}^{N}(y_\star - x_\ell)\bar{\kappa}(x_\ell, y_\star) = 0$, implying

$$y_\star = \mathbf{\Psi}_{\text{MS}}(y_\star) := \frac{\sum_{\ell=1}^{N} x_\ell\, \bar{\kappa}(x_\ell, y_\star)}{\sum_{\ell=1}^{N} \bar{\kappa}(x_\ell, y_\star)}. \tag{13}$$

The mean shift algorithm seeks the location of $y_\star$ using a fixed-point iteration on map $\mathbf{\Psi}_{\text{MS}}$. Convergence of mean shift is extensively investigated in (Cheng, 1995; Li et al., 2007; Carreira-Perpinan, 2007; Ghassabeh, 2013, 2015; Arias-Castro et al., 2016; Yamasaki and Tanaka, 2019, 2024). Assumption 2.1 is satisfied by **many distance-based kernels** defined as $\kappa(x, y) = \varphi(\|x-y\|)$ for some positive function $\varphi \in \mathcal{C}^1(\mathbb{R}^+)$. This includes the squared exponential kernel, the inverse multiquadric kernel (IMQ), and differentiable Matérn kernels.

2.3 Gradient flows for MMD minimization

A gradient flow of a functional $\mathcal{F}$ on the space of positive measures $\mathcal{M}_+(\mathcal{X})$ is a trajectory μ_t satisfying the partial differential equation (PDE)

$$\dot{\mu}_t = -\text{grad}_D \mathcal{F}[\mu_t], \tag{14}$$

where $\text{grad}_D \mathcal{F}$ denotes the gradient of $\mathcal{F}$ with respect to a metric $D : \mathcal{M}_+(\mathcal{X}) \times \mathcal{M}_+(\mathcal{X}) \to \mathbb{R}$; see (Otto, 2001; Ambrosio et al., 2008; Santambrogio, 2017; Chewi et al., 2024) for details. Gradient flows are extensively used in machine learning and statistics, particularly for sampling (Jordan et al., 1998; Craig and Bertozzi, 2016; Carrillo et al., 2019; Chewi et al., 2020; Salim et al., 2020; Glaser et al., 2021; Korba et al., 2021; Lu et al., 2023; Maurais and Marzouk, 2024).

For MMD minimization, the functional $\mathcal{F}$ and its first variation $\mathcal{F}_\delta$ are taken to be

$$\mathcal{F}(\mu) := \frac{1}{2}\text{MMD}(\mu, \pi)^2, \tag{15a}$$

$$\mathcal{F}_\delta[\mu](\cdot) = \int_{\mathcal{X}} \kappa(\cdot, x)\,\mathrm{d}\mu(x) - \int_{\mathcal{X}} \kappa(\cdot, x)\,\mathrm{d}\pi(x). \tag{15b}$$

The functional $\mathcal{F}_\delta$ quantifies how $\mathcal{F}$ changes due to a perturbation $\chi \in \mathcal{P}(\mathcal{X})$ around a measure μ, and is an essential object in the gradient flows literature. Table 1 summarizes various choices of metric D that have been used for MMD gradient flows, along with the corresponding PDE. We horizontally align corresponding terms

Here, we specifically consider literature that discretizes the dynamics of μ_t in (14) using interacting particles, but also include the MMD[1] and Fisher–Rao geometries for completeness; with this mean-field ODE perspective, however, these latter two geometries have limited use. Arbel et al. (2019) study the gradient flow of $\mathcal{F}$ under the 2-Wasserstein (W_2) geometry, while (Gladin et al., 2024) proposes inheriting properties from both the W_2 and MMD geometries via *spherical interaction-force transport* (IFT). Despite being able to use interacting particle systems, both W_2 and IFT flows of the MMD suffer from significantly degraded performance when initialized poorly. In particular, they require a large number of particles to converge, making the resulting quantization inefficient for, e.g., quadrature.

---

[1] Not to be confused with when $\mathcal{F}$ is the MMD.

Table 1: Gradient Flow Geometries for Minimizing MMD Functional $\mathcal{F}$

Geometry metric D	**PDE** $\dot{\mu}_t = -\mathrm{grad}_D \mathcal{F}[\mu_t]$			
Wasserstein (Arbel et al., 2019; Altekrüger et al., 2023)	$\dot{\mu}_t =$	$\alpha \text{ div}\left(\mu_t \nabla \mathcal{F}_\delta[\mu_t]\right)$		
Fisher–Rao	$\dot{\mu}_t =$		$-\beta\, \mathcal{F}_\delta[\mu_t]\mu_t$	
MMD	$\dot{\mu}_t =$			$-\gamma(\mu_t - \pi)$
IFT (Gladin et al., 2024; Zhu and Mielke, 2024)	$\dot{\mu}_t =$	$\alpha \text{ div}\left(\mu_t \nabla \mathcal{F}_\delta[\mu_t]\right)$		$-\gamma(\mu_t - \pi)$
WFR (Gladin et al., 2024)	$\dot{\mu}_t =$	$\alpha \text{ div}\left(\mu_t \nabla \mathcal{F}_\delta[\mu_t]\right)$	$-\beta\, \mathcal{F}_\delta[\mu_t]\mu_t$	

Finally, the Wasserstein-Fisher–Rao (WFR) geometry allows both mass transport and total mass variation, extending the W_2 distance to measures with different total masses, i.e., *unbalanced* optimal transport (Kondratyev et al., 2016; Gallouët and Monsaingeon, 2017; Chizat et al., 2018; Liero et al., 2018).[2] Compared to the Wasserstein gradient flow, the WFR gradient flow introduces a reaction term, which allows the total mass of μ_t to change as a function of t. Despite this formulation, the numerical scheme proposed by (Gladin et al., 2024) to solve this PDE artificially conserves mass by projecting the weights onto the simplex, inconsistent with the PDE's theoretical properties.

3 MAIN RESULTS

This section is structured as follows: In Section 3.1, we discretize a WFR gradient flow using a system of ODEs describing an interacting particle system (IPS), then show that the MMD decreases along the ODE solution. Moreover, we characterize conditions for the steady-state solution of these ODEs. In Section 3.2, we derive *mean shift interacting particles* (MSIP) as a fixed-point iteration for satisfying steady-state conditions of the WFR gradient flow. Then, we show that MSIP can be expressed as a preconditioned gradient descent on the MMD as a natural extension of the mean shift algorithm. Algorithms 1 and 2 in Section A.1 describe implementations of WFR-IPS and MSIP.

3.1 An IPS approach to WFR gradient flows

We consider the WFR gradient flow for minimizing the MMD (cf. Table 1), setting the reaction speed to $\beta \equiv 1$ without loss of generality:

$$\dot{\mu}_t = \alpha \text{ div}\left(\mu_t \nabla \mathcal{F}_\delta[\mu_t]\right) - \mathcal{F}_\delta[\mu_t]\mu_t. \tag{16}$$

We describe μ_t via M Diracs, i.e., $\mu_t = \sum_{i=1}^{M} w_i^{(t)} \delta_{y_i^{(t)}}$, for $y_i^{(t)} \in \mathcal{X}$ and $w_i^{(t)} \in \mathbb{R}$.

Proposition 3.1. *Define the system of ordinary differential equations*

$$\dot{y}_i^{(t)} = -\alpha \nabla \mathcal{F}_\delta[\mu_t](y_i^{(t)}),\ \dot{w}_i^{(t)} = -w_i^{(t)} \mathcal{F}_\delta[\mu_t](y_i^{(t)})$$

$$\mathcal{F}_\delta[\mu_t](y) = \sum_{m=1}^{M} w_m^{(t)} \kappa(y_m^{(t)}, y) - v_0(y) \tag{17}$$

where $i \in [M]$, $\alpha > 0$, and v_0 is defined in (5)*. If $(\mu_t)_{t\geq 0}$ solves* (17)*, then it weakly satisfies* (16)*.*

We can then use standard ODE solvers to efficiently approximate μ_t, provided the function v_0 can be evaluated and differentiated. When π is an empirical measure of N atoms in $\mathbb{R}^d$, this simply requires differentiability of kernel κ, and simulating (17) is $O(d(M+N)M)$ operations.[3] The proof and a more detailed account of Proposition 3.1 are given in Section D.1.

Proposition 3.2. *Assume that $\kappa \in \mathcal{C}^1$ has bounded gradient, $\kappa(x,y) \geq 0$ for any $x, y \in \mathcal{X}$, and $y \mapsto \kappa(y,y)$ is a constant function with value $B_\kappa > 0$. Let $y_i \in \mathcal{X}$ and $w_i \in (0,1)$ for $i \in [M]$. Then there exists a unique solution of* (17) *over $t \in [0, +\infty)$ with $y_i^{(0)} = y_i$ and $w_i^{(0)} = w_i$. Further, we have $w_i^{(t)} \in (0,1)$ for $t \geq 0$, and $t_2 \geq t_1 \geq 0$ implies*

$$\mathcal{F}(\mu_{t_2}) \leq \mathcal{F}(\mu_{t_1}). \tag{18}$$

The proof of Proposition 3.2 is given in Section D.2. The system of ODEs (17) yields quadrature rules with bounded non-negative weights, which is a desirable property in the context of numerical integration (Karvonen et al., 2019). As $\mathcal{F}$ is bounded from below, we know that (17) approaches a stationary point as $t \to \infty$. We are then primarily interested in the minimization of the MMD when the particles and weights reach stationarity as well, i.e., $\dot{y}_i^{(t)} = 0$ and $\dot{w}_i^{(t)} = 0$ as $t \to \infty$, which is not necessarily guaranteed by Proposition 3.2. Conditions on y_i and w_i sufficient for stationarity are:

$$v_0(y_i) = \sum_{m=1}^{M} w_m \kappa(y_m, y_i), \tag{19}$$

---

[2] While we use the convention from (Gallouët and Monsaingeon, 2017), Mielke and Zhu (2025, Remark 2.2) address the historical differentiation of the Fisher–Rao and closely-related Hellenger geometries.

[3] This assumes evaluating κ is $O(1)$ and $\nabla_2 \kappa$ is $O(d)$.

$$\nabla v_0(y_i) = \sum_{m=1}^{M} w_m \nabla_2 \kappa(y_m, y_i). \tag{20}$$

Condition (19) is prevalent in the literature on kernel-based quadrature, and reflects that the quadrature rule defined by the measure $\mu = \sum_{m=1}^{M} w_m \delta_{y_m}$ is exact on the subspace spanned by the functions $\kappa(\cdot, y_i)$, equivalent to (4). Above, we have made few assumptions on the structure of κ. We now use Assumption 2.1 to explore condition (20), less known in the literature.

Proposition 3.3. *Let $Y \in \mathcal{X}^M$. Under Assumption 2.1 and assuming that the gradient of κ is bounded on $\mathcal{X}$,* (19) *and* (20) *imply that*

$$\bar{\boldsymbol{K}}(Y)\boldsymbol{W}(Y)Y = \hat{\boldsymbol{v}}_1(Y), \tag{21}$$

where $\bar{\boldsymbol{K}}(Y) \in \mathbb{R}^{M\times M}$ is the kernel matrix with kernel $\bar{\kappa}$, the diagonal matrix $\boldsymbol{W}(Y)$ has entries corresponding to those of $\hat{\boldsymbol{w}}(Y)$ given by (4)*, and $\hat{\boldsymbol{v}}_1(Y) \in \mathbb{R}^{M\times d}$ is the matrix with rows defined as*

$$(\hat{\boldsymbol{v}}_1(Y))_{i,:} := \nabla v_0(y_i) + y_i \sum_{m=1}^{M} \hat{w}_m \bar{\kappa}(y_m, y_i). \tag{22}$$

When using the squared-exponential kernel with unit bandwidth, (19), (20), and (22) give

$$(\hat{\boldsymbol{v}}_1(Y))_{i,:} = v_1(y_i) := \int x\kappa(x, y_i)\,\mathrm{d}\pi(x). \tag{23}$$

Thus v_1 resembles a “kernelized first moment” (hence the subscript), similar to how $v_0(y_i)$, the KDE of π, resembles a “kernelized zeroth moment.” The proof of Proposition 3.3 is given in Section D.3.

3.2 A fixed-point scheme for steady-state solutions

In general, we cannot analytically find Y satisfying (21). Instead, consider the sequence $(Y^{(t)})$ with

$$\bar{\boldsymbol{K}}(Y^{(t)})\boldsymbol{W}(Y^{(t)})Y^{(t+1)} = \hat{\boldsymbol{v}}_1(Y^{(t)}). \tag{24}$$

Calculating $\boldsymbol{W}(Y^{(t)})$ of course requires the invertibility of $\boldsymbol{K}(Y^{(t)})$. If $\bar{\boldsymbol{K}}(Y^{(t)})$ is also nonsingular, (24) is a fixed-point iteration, $Y^{(t+1)} = \boldsymbol{\Psi}_{\mathrm{MSIP}}(Y^{(t)})$, with

$$\boldsymbol{\Psi}_{\mathrm{MSIP}}(Y) := \boldsymbol{W}(Y)^{-1}\bar{\boldsymbol{K}}(Y)^{-1}\hat{\boldsymbol{v}}_1(Y). \tag{25}$$

We call $\boldsymbol{\Psi}_{\mathrm{MSIP}}$ the *mean shift interacting particles* map. We now show that iteration of Ψ_{MSIP} can be seen as a preconditioned gradient descent of the function $F_M : \mathcal{X}^M \to \mathbb{R}$ given by

$$F_M(y_1, \ldots, y_M) := \inf_{\boldsymbol{w}\in\mathbb{R}^M} \mathcal{F}\left[\sum_{i=1}^{M} w_i \delta_{y_i}\right]. \tag{26}$$

Theorem 3.4. *Let $Y \in \mathcal{X}^M$ be such that* $\det \boldsymbol{K}(Y) > 0$ *and define $\boldsymbol{P}(Y) := \boldsymbol{W}(Y)\bar{\boldsymbol{K}}(Y)\boldsymbol{W}(Y)$. Under Assumption 2.1, we have*

$$\begin{aligned}\nabla F_M(Y) &= \boldsymbol{W}(Y)\left(\bar{\boldsymbol{K}}(Y)\boldsymbol{W}(Y)Y - \hat{\boldsymbol{v}}_1(Y)\right)\\ &= \boldsymbol{P}(Y)\Big(Y - \boldsymbol{\Psi}_{\mathrm{MSIP}}(Y)\Big)\end{aligned} \tag{27}$$

with $\nabla F_M : \mathcal{X}^M \to \mathbb{R}^{M\times d}$ and $\boldsymbol{\Psi}_{\mathrm{MSIP}}(Y)$ from (25) *when $\boldsymbol{P}(Y)$ is invertible.*

Theorem 3.4 is proved in Section D.4. The identity (27) proves that the fixed points of the function $\boldsymbol{\Psi}_{\mathrm{MSIP}}$ are critical points of the function F_M. This result is reminiscent of (9), proved in (Du et al., 1999), which gives a similar characterization of the fixed points of Lloyd’s map $\boldsymbol{\Psi}_{\mathrm{Lloyd}}$. Following identical reasoning as (Portales et al., 2025), we deduce from (27) that the fixed point iteration defined by (25) is a preconditioned gradient descent, with $\boldsymbol{P}(Y)$ as a preconditioning matrix. If matrix $\boldsymbol{P}(Y^{(t)})$ is nonsingular, (25) and (27) yield

$$Y^{(t+1)} = Y^{(t)} - \boldsymbol{P}(Y^{(t)})^{-1}\nabla F_M(Y^{(t)}). \tag{28}$$

If we choose a step size η for (28), we obtain a damped fixed-point iteration (Kelley, 2018) on $\boldsymbol{\Psi}_{\mathrm{MSIP}}$ as

$$\begin{aligned}Y^{(t+1)} &= Y^{(t)} - \eta\boldsymbol{P}(Y^{(t)})^{-1}\nabla F_M(Y^{(t)})\\ &= (1-\eta)Y^{(t)} + \eta\boldsymbol{\Psi}_{\mathrm{MSIP}}(Y^{(t)}).\end{aligned} \tag{29}$$

In the case of $\kappa \propto \bar{\kappa}$, i.e., squared-exponential (Cheng, 1995), we can use $\boldsymbol{v}_1$ from (23) to simplify (29) into

$$Y^{(t+1)} = (1-\eta)Y^{(t)} + \eta\boldsymbol{W}(Y^{(t)})^{-1}\boldsymbol{K}(Y^{(t)})^{-1}\boldsymbol{v}_1(Y^{(t)}). \tag{30}$$

The following result shows the existence of a step size schedule for which the MMD decreases along the iterates of (30).

Proposition 3.5. *Let κ be a squared-exponential kernel. For any $Y^{(0)}$ such that $\hat{\boldsymbol{w}}(Y^{(0)})$ exists and has strictly positive entries, there exist positive step sizes $(\eta_t)_{t\geq 0}$ such that the sequence*

$$Y^{(t+1)} = (1-\eta_t)Y^{(t)} + \eta_t\boldsymbol{W}(Y^{(t)})^{-1}\boldsymbol{K}(Y^{(t)})^{-1}\boldsymbol{v}_1(Y^{(t)}) \tag{31}$$

satisfies $F_M(Y^{(t+1)}) \leq F_M(Y^{(t)})$ for $t \geq 0$, with invertible $\boldsymbol{K}(Y^{(t)})$ and positive $\hat{\boldsymbol{w}}(Y^{(t)})$.

Proposition 3.6. *Under the assumptions of Proposition 3.5 and for unweighted empirical π, if $Y^{(0)}$ are distinct points on the support of π, then there exists a bandwidth of κ for which $\hat{\boldsymbol{w}}(Y^{(0)}) > 0$.*

The proofs of Propositions 3.5 and 3.6 are given in Sections D.5 and D.6. Proposition 3.5 yields a descent property for (29), allowing for varying step sizes.

Propositions 3.5 and 3.6 resemble early existence and convergence guarantees for Lloyd's algorithm, refined over many decades (Du et al., 2006; Pagès and Yu, 2016; Portales et al., 2025). Further, the bandwidth consideration in Proposition 3.6 is common in kernel methods (Chiu, 1991). While Proposition 3.5 requires initially positive weights for theoretical purposes, we have observed empirical success when initializing a configuration with negative weights. Extending Propositions 3.5 and 3.6 to arbitrary initialization and fixed step size is left to future work: guaranteeing convergence for a fixed step size requires higher-order analysis of the function F_M and is beyond the scope of this work.

3.3 Mean shift as an accelerated gradient flow

In the case of one particle, i.e., $M = 1$, we show that the classical mean shift algorithm identifies critical points of the function F_1 given in (26).

Corollary 3.7. *When π is an empirical distribution, under Assumption 2.1 and further assuming $\bar{v}_0(y) := \int_{\mathcal{X}} \bar{\kappa}(x, y)\,\mathrm{d}\pi(x) \neq 0$ and $v_0(y) \neq 0$, the following expressions are equal*

$$\nabla F_1(y) \equiv \frac{v_0^2(y)\bar{\kappa}(y,y)}{\kappa^2(y,y)}(y - \mathbf{\Psi}_{\mathrm{MSIP}}(y)) = \\ \equiv \frac{v_0(y)\bar{v}_0(y)}{\kappa(y,y)}(y - \mathbf{\Psi}_{\mathrm{MS}}(y)), \tag{32}$$

where $\mathbf{\Psi}_{\mathrm{MS}}$ is defined by (13). *In particular, when $\bar{\kappa} = \lambda\kappa$ for $\lambda \neq 0$, $\mathbf{\Psi}_{\mathrm{MSIP}}(y) = \mathbf{\Psi}_{\mathrm{MS}}(y)$.*

Corollary 3.7 shows the intimate connections between the mean shift interacting particles and classical mean shift maps. In the simple case of squared-exponential κ and a single particle, the maps are the same. More broadly, for any critical point y of F_1, satisfying the assumptions of Corollary 3.7 automatically yields that y is a critical point of *both* $\mathbf{\Psi}_{\mathrm{MSIP}}$ and $\mathbf{\Psi}_{\mathrm{MS}}$. The proof of Corollary 3.7 is given in Section D.7 as a direct consequence of Theorem 3.4.

In this single particle setting, we contrast the definition of MSIP (30) with an unweighted Wasserstein gradient flow for the MMD (Arbel et al., 2019). For simplicity, we consider squared-exponential κ with unit bandwidth, unifying MSIP and mean shift. The algorithm derived in (Arbel et al., 2019) boils down to the iteration

$$y^{(t+1)} = y^{(t)} + \eta\nabla v_0(y^{(t)}) \\ = \big(1 - \eta v_0(y^{(t)})\big)y^{(t)} + \eta v_1(y^{(t)}). \tag{33}$$

When $y^{(0)}$ is far from the data, small $v_0(y^{(t)})$ and $\|v_1(y^{(t)})\|$ give practically still particles: $y^{(1)} \approx y^{(0)}$. The preconditioner in (30), however, prevents mean shift steps from vanishing in unlikely regions. Thus, we expect mean-shift to **accelerate steady-state behavior** over typical gradient-flow methods. Section A.5 and Figure 6 give additional details on this phenomenon.

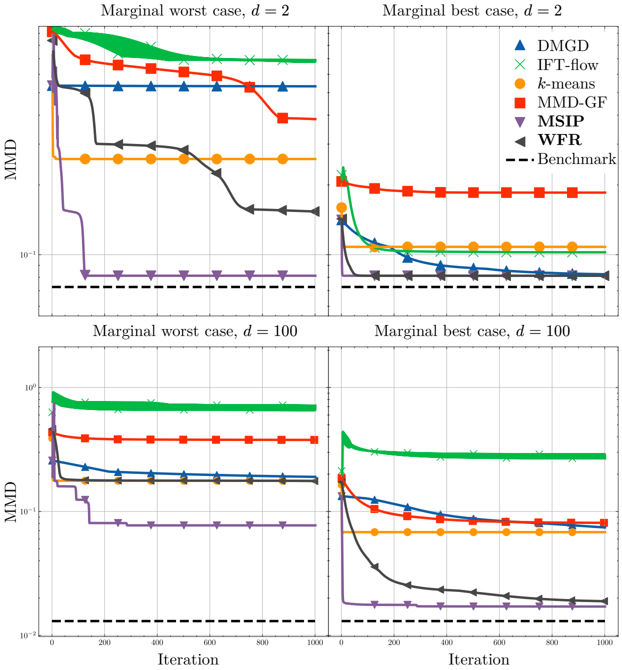


Figure 2: Comparison of different quantization algorithms on a GMM. (Top): dimension $d = 2$, $L_0 = M = 3$. (Bottom): dimension $d = 100$, $L_0 = 5$, $M = 10$. We use squared-exponential kernel with $\sigma = 5$ for kernelized algorithms with hyperparameter tuning.

4 NUMERICAL RESULTS

We now summarize numerical simulations that validate the proposed algorithms.[4] In Section 4.1 we conduct numerical experiments on Gaussian mixture datasets, while in Section 4.2, we perform the experiments on MNIST. We provide additional details about these numerics in Sections A.2 to A.4, comparisons to kernel thinning (Dwivedi and Mackey, 2024) in Section A.7, examples for many distributions in Section A.8, and resulting weights of our algorithms in Section A.9. We demonstrate further practicality for data science by clustering high-dimensional features of cellular images in Section B.

4.1 Experiment 1: Gaussian mixture datasets

We consider quantization of two Gaussian mixture distributions, represented as empirical measures of $N = 1000$ atoms: a mixture of $L_0 = 3$ Gaussians in $d = 2$ and a mixture of $L_0 = 5$ Gaussians in $d = 100$.

---

[4] github.com/AyoubBelhadji/disruptive_quantization

We compare the following algorithms: i) Wasserstein–Fisher–Rao Flow (WFR) from Section 3.1, ii) mean shift interacting particles (MSIP) from Section 3.2, iii) IFTflow (Gladin et al., 2024), iv) MMD gradient flow (MMDGF) (Arbel et al., 2019), v) k-means, vi) discrepancy-minimizing gradient descent (DMGD) on the function F_M in (26) without preconditioning. Corollary 3.7 suggests MSIP will outperform DMGD.

Figure 2 shows MMD as a function of iteration number for quantizations obtained from running each algorithm on 100 random initializations. At each iteration and for each algorithm, we plot the highest and lowest MMD value across all 100 initializations. We observe that, even in the worst case, MSIP consistently outperforms other quantization algorithms, quickly converging to a near-optimal quantization under the MMD. The latter is captured by the dashed line, which is a standard benchmark for the MMD of optimal quadrature with M points; see details in Section A.2. For $d = 2$, WFR more slowly optimizes the MMD, eventually surpassing k-means. In the best case, all algorithms perform well except for MMDGF. When $d = 100$, we have a more difficult approximation problem, with MSIP performing best in both the best- and worst-case instances at each iteration. WFR performs comparably to k-means in the worst case and to MSIP in the best case. Other algorithms struggle in this high-dimensional problem.

Qualitatively, Figure 5 in Section A.4 further illustrates the sensitivity of each algorithm to its initialization. MMDGF, IFTFlow, and DMGD are not robust to initializations in the two-dimensional example, whereas MSIP and WFR perform well even when initialized far from the target support; k-means lies in between.

4.2 Experiment 2: MNIST

We now illustrate our algorithms using the MNIST dataset (LeCun et al., 2010); for further results, see Section A.6.2. We compare MSIP, Lloyd's algorithm, WFR, IFTflow, MMDGF, DMGD, and classical (non-interacting) mean shift (IIDMS). When Lloyd's algorithm produces an empty Voronoi cell, we make the corresponding particle retain its position.

Figure 3 shows the results of 5000 iterations for each algorithm using a Matérn kernel, all initialized with the same set of M i.i.d. samples drawn uniformly from $[0,1]^{784}$. We see that MSIP and WFR recover recognizable digits. MSIP and WFR each output two particles that look like ones: these image pairs correspond to different modes of MNIST, distant in pixel-space. IIDMS collapses to a single mode, showing no diversity whatsoever. Lloyd's algorithm has difficulty using all centroids, as several particles have empty Voronoi cells across all iterations[5]; the cells of the remaining particles cover many digits. The other three algorithms remain trapped far from the distribution's support. This experiment illustrates the mode-seeking behavior of our algorithms for multi-modal distributions and initialization robustness in high-dimensional domains.

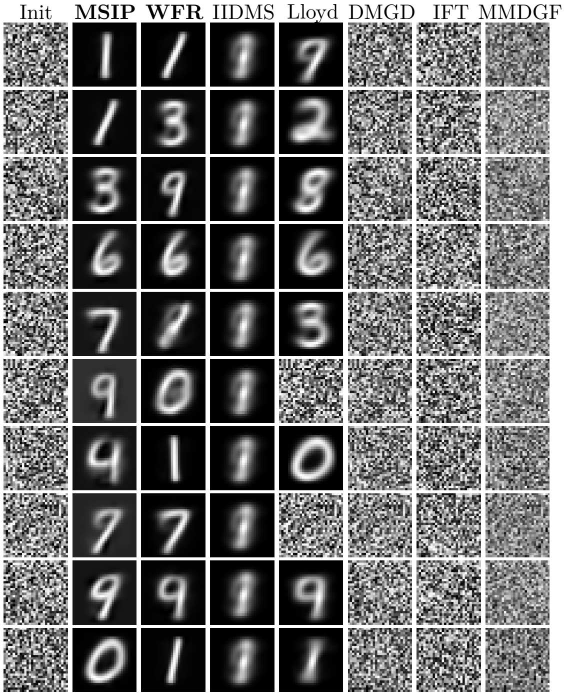


Figure 3: Comparing quantizations of MNIST

5 DISCUSSION

This work suggests further theoretical developments for MMD minimization that bypass mean-field analysis. In particular, the unifying perspective developed here connects, for the first time, MMD minimization to convergence theory for Lloyd's algorithm and the mean shift algorithm. We expect that this connection will provide new tools for analyzing the difficult non-convex optimization problems intrinsic to quantization. We have already characterized, e.g., via Proposition 3.2 and Theorem 3.4, non-trivial properties of WFR-IPS and MSIP. Future work could investigate possible Łojasiewicz properties for our algorithms; indeed, Kurdyka–Łojasiewicz inequalities were recently used to establish theoretical guarantees for both Lloyd's algorithm and mean shift (Portales et al., 2025; Yamasaki and Tanaka, 2024). Beyond convergence guarantees, an interesting direction for future research is to explore the impact of weighted quantization on the understanding of neural network training (Arbel et al., 2019; Rotskoff et al., 2019) and on

---

[5] Lloyd's algorithm is typically initialized from the data distribution itself to mitigate this phenomenon.

sparse measure reconstructions (De Castro and Gamboa, 2012; Chizat, 2022; Belhadji and Gribonval, 2024). We also believe that MSIP and WFR-IPS can be extended for sampling from unnormalized densities (Liu and Wang, 2016; Chen et al., 2018; Korba et al., 2021; Maurais and Marzouk, 2024). Generally, our methods show promising results for mode seeking using interacting particle systems, indicating that such systems are well suited to general problems of non-convex high-dimensional distribution approximation.

Acknowledgments

AB, DS, and YM acknowledge support from the US Department of Energy (DOE), Office of Advanced Scientific Computing Research, under grants DE-SC0021226 (FASTMath SciDAC Institute) and DE-SC0023188. AB and YM also acknowledge support from the ExxonMobil Technology and Engineering Company. DS also acknowledges support from a 2025–26 MathWorks Engineering Fellowship at MIT. The authors also acknowledge the MIT Office of Research Computing and Data, MIT SuperCloud, and Lincoln Laboratory Supercomputing Center for providing HPC resources that have contributed to the research results reported within this paper.

References

Acevedo, A., Alférez, S., Merino, A., Puigví, L., and Rodellar, J. (2019). Recognition of peripheral blood cell images using convolutional neural networks. *Computer Methods and Programs in Biomedicine*, 180:105020.

Altekrüger, F., Hertrich, J., and Steidl, G. (2023). Neural Wasserstein gradient flows for maximum mean discrepancies with Riesz kernels. *ICML*.

Ambrosio, L., Gigli, N., and Savaré, G. (2008). *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media.

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

Arias-Castro, E., Mason, D., and Pelletier, B. (2016). On the estimation of the gradient lines of a density and the consistency of the mean-shift algorithm. *Journal of Machine Learning Research*, 17(206):1–4.

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

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

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

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

Belhadji, A. and Gribonval, R. (2024). Sketch and shift: a robust decoder for compressive clustering. *Transactions on Machine Learning Research*.

Briol, F.-X., Oates, C., Girolami, M., and Osborne, M. A. (2015). Frank-Wolfe Bayesian quadrature: Probabilistic integration with theoretical guarantees. *Advances in Neural Information Processing Systems*, 28.

Carreira-Perpinan, M. A. (2007). Gaussian mean-shift is an EM algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):767–776.

Carreira-Perpinán, M. A. (2015). A review of mean-shift algorithms for clustering. *arXiv preprint*.

Carrillo, J. A., Craig, K., and Patacchini, F. S. (2019). A blob method for diffusion. *Calculus of Variations and Partial Differential Equations*, 58:1–53.

Chatalic, A., Schreuder, N., De Vito, E., and Rosasco, L. (2023). Efficient numerical integration in reproducing kernel Hilbert spaces via leverage scores sampling. *arXiv preprint*.

Chen, W., Mackey, L., Gorham, J., Briol, F., and Oates, C. (2018). Stein points. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 844–853. PMLR.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799.

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

Chewi, S., Niles-Weed, J., and Rigollet, P. (2024). Statistical optimal transport. *arXiv preprint*.

Chiu, S.-T. (1991). Bandwidth selection for kernel density estimation. *The Annals of Statistics*, pages 1883–1905.

Chizat, L. (2022). Sparse optimization on measures with over-parameterized gradient descent. *Mathematical Programming*, 194(1):487–532.

Chizat, L., Oyallon, E., and Bach, F. (2019). On lazy training in differentiable programming. *Advances in neural information processing systems*, 32.

Chizat, L., Peyré, G., Schmitzer, B., and Vialard, F. (2018). An interpolating distance between optimal transport and Fisher–Rao metrics. *Foundations of Computational Mathematics*, 18:1–44.

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619.

Craig, K. and Bertozzi, A. (2016). A blob method for the aggregation equation. *Mathematics of computation*, 85(300):1681–1717.

De Castro, Y. and Gamboa, F. (2012). Exact reconstruction using beurling minimal extrapolation. *Journal of Mathematical Analysis and applications*, 395(1):336–354.

De M., S. (2003). On optimal center locations for radial basis function interpolation: computational aspects. *Rend. Splines Radial Basis Functions and Applications*, 61(3):343–358.

De M., S., Schaback, R., and Wendland, H. (2005). Near-optimal data-independent point locations for radial basis function interpolation. *Advances in Computational Mathematics*, 23:317–330.

Du, Q., Emelianenko, M., and Ju, L. (2006). Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations. *SIAM journal on numerical analysis*, 44(1):102–119.

Du, Q., Faber, V., and Gunzburger, M. (1999). Centroidal Voronoi tessellations: Applications and algorithms. *SIAM review*, 41(4):637–676.

Dwivedi, R. and Mackey, L. (2022). Generalized kernel thinning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. PMLR.

Dwivedi, R. and Mackey, L. (2024). Kernel thinning. *Journal of Machine Learning Research*, 25(152):1–77.

Dziugaite, G. K., Roy, D. M., and Ghahramani, Z. (2015). Training generative neural networks via maximum mean discrepancy optimization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, page 258–267.

Ehler, M., Gräf, M., and Oates, C. J. (2019). Optimal Monte Carlo integration on closed manifolds. *Statistics and Computing*, 29(6):1203–1214.

Emelianenko, M., Ju, L., and Rand, A. (2008). Nondegeneracy and weak global convergence of the Lloyd algorithm in $\mathbb{R}^d$. *SIAM Journal on Numerical Analysis*, 46(3):1423–1441.

Epperly, E. and Moreno, E. (2023). Kernel quadrature with randomly pivoted Cholesky. *Advances in Neural Information Processing Systems*, 36:65850–65868.

Fukunaga, K. and Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1):32–40.

Gallouët, T. O. and Monsaingeon, L. (2017). A JKO splitting scheme for Kantorovich–Fisher–Rao gradient flows. *SIAM Journal on Mathematical Analysis*, 49(2):1100–1130.

Ghassabeh, Y. A. (2013). On the convergence of the mean shift algorithm in the one-dimensional space. *Pattern Recognition Letters*, 34(12):1423–1427.

Ghassabeh, Y. A. (2015). A sufficient condition for the convergence of the mean shift algorithm with Gaussian kernel. *Journal of Multivariate Analysis*, 135:1–10.

Gladin, E., Dvurechensky, P., Mielke, A., and Zhu, J.-J. (2024). Interaction-force transport gradient flows. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C., editors, *Advances in Neural Information Processing Systems*, volume 37, pages 14484–14508. Curran Associates, Inc.

Glaser, P., Arbel, M., and Gretton, A. (2021). KALE flow: A relaxed KL gradient flow for probabilities with disjoint support. *Advances in Neural Information Processing Systems*, 34:8018–8031.

Graf, S. and Luschgy, H. (2000). *Foundations of quantization for probability distributions*. Springer Science & Business Media.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

Gretton, A., Fukumizu, K., Teo, C., Song, L., Schölkopf, B., and Smola, A. (2007). A kernel statistical test of independence. *Advances in neural information processing systems*, 20.

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

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

Hertrich, J., Wald, C., Altekrüger, F., and Hagemann, P. (2024). Generative sliced MMD flows with Riesz kernels. In *The Twelfth International Conference on Learning Representations*.

Huszár, F. and Duvenaud, D. (2012). Optimally-weighted herding is bayesian quadrature. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, UAI'12, page 377–386. AUAI Press.

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

Karvonen, T., Kanagawa, M., and Särkkä, S. (2019). On the positivity and magnitudes of Bayesian quadrature weights. *Statistics and Computing*, 29:1317–1333.

Karvonen, T. and Särkkä, S. (2018). Fully symmetric kernel quadrature. *SIAM Journal on Scientific Computing*, 40(2):A697–A720.

Karvonen, T. and Särkkä, S. (2019). Gaussian kernel quadrature at scaled Gauss–Hermite nodes. *BIT Numerical Mathematics*, 59(4):877–902.

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

Kelley, C. T. (2018). Numerical methods for nonlinear equations. *Acta Numerica*, 27:207–287.

Kieffer, J. (1982). Exponential rate of convergence for Lloyd's method I. *IEEE Transactions on Information Theory*, 28(2):205–210.

Kondratyev, S., Monsaingeon, L., and Vorotnikov, D. (2016). A new optimal transport distance on the space of finite Radon measures. *Advances in Differential Equations*, 21(11/12).

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

Lacoste-Julien, S., Lindsten, F., and Bach, F. (2015). Sequential kernel herding: Frank-Wolfe optimization for particle filtering. In *Artificial Intelligence and Statistics*, pages 544–552. PMLR.

Lam, S. K., Pitrou, A., and Seibert, S. (2015). Numba: A LLVM-based Python JIT compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6.

LeCun, Y., Cortes, C., and Burges, C. (2010). MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Li, C.-L., Chang, W.-C., Cheng, Y., Yang, Y., and Póczos, B. (2017). MMD GAN: Towards deeper understanding of moment matching network. *Advances in neural information processing systems*, 30.

Li, X., Hu, Z., and Wu, F. (2007). A note on the convergence of the mean shift. *Pattern recognition*, 40(6):1756–1762.

Liero, M., Mielke, A., and Savaré, G. (2018). Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures. *Inventiones mathematicae*, 211(3):969–1117.

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

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137.

Lu, Y., Lu, J., and Nolen, J. (2019). Accelerating Langevin sampling with birth-death. *arXiv preprint*.

Lu, Y., Slepčev, D., and Wang, L. (2023). Birth–death dynamics for sampling: global convergence, approximations and their asymptotics. *Nonlinearity*, 36(11):5731.

Maurais, A. and Marzouk, Y. (2024). 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.

Mielke, A. and Zhu, J.-J. (2025). Hellinger-kantorovich gradient flows: Global exponential decay of entropy functionals.

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

Neal, R. M. (2003). Slice sampling. *The Annals of Statistics*, 31(3).

Oettershagen, J. (2017). *Construction of optimal cubature algorithms with applications to econometrics and uncertainty quantification.* Verlag Dr. Hut.

Otto, F. (2001). The geometry of dissipative evolution equations: the porous medium equation. *Communications in Partial Differential Equations.*

Owen, A. B. (2017). Statistically efficient thinning of a Markov chain sampler. *Journal of Computational and Graphical Statistics*, 26(3):738–744.

Pagès, G. and Yu, J. (2016). Pointwise convergence of the Lloyd algorithm in higher dimension. *SIAM Journal on Control and Optimization*, 54(5):2354–2382.

Peyré, G. and Cuturi, M. (2019). Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Pinkus, A. (2012). *n-Widths in Approximation Theory*, volume 7. Springer Science & Business Media.

Portales, L., Cazelles, E., and Pauwels, E. (2025). On the Sequential Convergence of Lloyd's Algorithms. *Mathematics of Operations Research*, page moor.2024.0550.

Riabiz, M., Chen, W. Y., Cockayne, J., Swietach, P., Niederer, S. A., Mackey, L., and Oates, C. J. (2022). Optimal thinning of MCMC output. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(4):1059–1081.

Robert, C. P., Casella, G., and Casella, G. (1999). *Monte Carlo statistical methods*, volume 2. Springer.

Rotskoff, G., Jelassi, S., Bruna, J., and Vanden-Eijnden, E. (2019). Global convergence of neuron birth-death dynamics. In *International Conference on Machine Learning*.

Salim, A., Korba, A., and Luise, G. (2020). The Wasserstein proximal gradient algorithm. *Advances in Neural Information Processing Systems*, 33:12356–12366.

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

Santin, G. and Haasdonk, B. (2017). Convergence rate of the data-independent P-greedy algorithm in kernel-based approximation. *Dolomites Research Notes on Approximation*, 10(06/2017):68–78.

Teschl, G. (2012). *Ordinary differential equations and dynamical systems*, volume 140. American Mathematical Soc.

Teymur, O., Gorham, J., Riabiz, M., and Oates, C. (2021). Optimal quantisation of probability measures using maximum mean discrepancy. In *International Conference on Artificial Intelligence and Statistics*, pages 1027–1035. PMLR.

Virtanen, P., Gommers, R., Oliphant, T., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S., Brett, M., Wilson, J., Millman, K., Mayorov, N., Nelson, A., Jones, E., Kern, R., Larson, E., Carey, C., Polat, İ., Feng, Y., Moore, E., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E., Harris, C., Archibald, A., Ribeiro, A., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Wu, X. (1992). On convergence of Lloyd's method I. *IEEE Transactions on Information Theory*, 38(1):171–174.

Yamasaki, R. and Tanaka, T. (2019). Properties of mean shift. *IEEE transactions on pattern analysis and machine intelligence*, 42(9):2273–2286.

Yamasaki, R. and Tanaka, T. (2024). Convergence analysis of mean shift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yan, Y., Wang, K., and Rigollet, P. (2024). Learning Gaussian mixtures using the Wasserstein–Fisher–Rao gradient flow. *The Annals of Statistics*, 52(4):1774–1795.

Zhu, J.-J. and Mielke, A. (2024). Kernel approximation of Fisher-Rao gradient flows. *arXiv preprint*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**, see linked repository.]
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**, we use Assumption 2.1 and any other assumptions are included in the individual result.]
   (b) Complete proofs of all theoretical results. [**Yes**, set Section D.]
   (c) Clear explanations of any assumptions. [**Yes**]
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**; see `experiment_configs` subdirectory in the repository.]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**.]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**: We describe any error bars below appropriate figures in the appendices.]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**: See Section A.2.]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [**Yes**]
   (b) The license information of the assets, if applicable. [**Not Applicable**]
   (c) New assets either in the supplemental material or as a URL, if applicable. [**Not Applicable**]
   (d) Information about consent from data providers/curators. [**Not Applicable**]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [**Not Applicable**]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [**Not Applicable**]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [**Not Applicable**]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [**Not Applicable**]

A Additional numerical details

In this section, we provide details on the algorithms, numerical simulations presented in Section 4, along with additional simulations.

A.1 Algorithm listings

Algorithm 1: Wasserstein–Fisher–Rao Interacting Particle System for MMD: Euler discretization

```
Input: Samples $x_1, x_2, \dots, x_N \sim \pi$, optimization iterations $T$, differentiable kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$, step size
       $\eta$, speed $\alpha \in \mathbb{R}^+$, original configuration $y_1^{(0)}, y_2^{(0)}, \dots, y_M^{(0)} \in \mathcal{X}$ and weights $w_1^{(0)}, w_2^{(0)}, \dots, w_M^{(0)} \in \mathbb{R}$
Output: Centroids $y_1^{(T)}, \dots, y_M^{(T)} \in \mathcal{X}$, weights $w_1^{(T)}, \dots, w_M^{(T)} \in \mathbb{R}$
Set $v_0(y) = \frac{1}{N} \sum_{j=1}^{N} \kappa(x_j, y)$
for $t = 0, 1, \dots, T-1$ do
    for $i = 1, 2, \dots, M$ do
        $y_i^{(t+1)} = y_i^{(t)} - \eta\,\alpha \left( \sum_{m=1}^{M} w_m^{(t)} \nabla_2 \kappa(y_m^{(t)}, y_i^{(t)}) - \nabla v_0(y_i^{(t)}) \right)$
        $w_i^{(t+1)} = w_i^{(t)} - \eta\, w_i^{(t)} \left( \sum_{m=1}^{M} w_m^{(t)} \kappa(y_m^{(t)}, y_i^{(t)}) - v_0(y_i^{(t)}) \right)$
    end
end
```

Algorithm 2: Mean shift interacting particles (MSIP)

```
Input: Samples $x_1, x_2, \dots, x_N \sim \pi$, optimization iterations $T$, differentiable kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$, kernel
       $\bar{\kappa}$ s.t. $\nabla_2 \kappa(x, y) = (x - y)\bar{\kappa}(x, y)$, step size $\eta$, original configuration $y_1^{(0)}, y_2^{(0)}, \dots, y_M^{(0)} \in \mathcal{X}$
Set $v_0(y) = \frac{1}{N} \sum_{j=1}^{N} \kappa(x_j, y)$
for $t = 0, 1, \dots, T-1$ do
    Calculate $\boldsymbol{K}(Y^{(t)})$ and $\bar{\boldsymbol{K}}(Y^{(t)})$ as $(\boldsymbol{K}(Y^{(t)}))_{i,m} = \kappa(y_i^{(t)}, y_m^{(t)})$ and $(\bar{\boldsymbol{K}}(Y^{(t)}))_{i,m} = \bar{\kappa}(y_i^{(t)}, y_m^{(t)})$
    Set $\boldsymbol{v}_0(Y^{(t)})$ as $(\boldsymbol{v}_0(Y^{(t)}))_i = v_0(y_i^{(t)})$ and $\hat{\boldsymbol{w}}(Y^{(t)}) = \boldsymbol{K}(Y^{(t)})^{-1} \boldsymbol{v}_0(Y^{(t)})$
    Set $\boldsymbol{W}(Y^{(t)}) = \mathrm{diag}(\hat{\boldsymbol{w}}(Y^{(t)}))$ and $\hat{\boldsymbol{v}}_1(Y^{(t)})$ such that
     $(\hat{\boldsymbol{v}}_1(Y^{(t)}))_{i,:} = \nabla v_0(y_i^{(t)}) + y_i^{(t)} \sum_{m=1}^{M} \hat{w}_m(Y^{(t)}) \bar{\kappa}(y_m^{(t)}, y_i^{(t)})$
    Calculate $Y^{(t+1)} = (1 - \eta) Y^{(t)} + \eta \boldsymbol{W}(Y^{(t)})^{-1} \bar{\boldsymbol{K}}(Y^{(t)})^{-1} \hat{\boldsymbol{v}}_1(Y^{(t)})$
end
Calculate $\hat{\boldsymbol{w}}(Y^{(T)}) = \boldsymbol{K}(Y^{(T)})^{-1} \boldsymbol{v}_0(Y^{(T)})$ with $\boldsymbol{K}(Y^{(T)})$ and $\boldsymbol{v}_0(Y^{(T)})$ as above
```

A.2 Implementation Details

First, we describe the experiment in Figure 1.

- We use ten particles drawn i.i.d. from a Gaussian over the top-right mode with mean $(0.98, 0.68)$ and covariance $\sigma^2 I$, and $\sigma^2 = 0.025$.
- WFR-IPS, MSIP, and mean shift use a squared exponential kernel with bandwidth 0.6.
- MSIP takes 1000 steps of size 0.5.
- WFR-IPS uses diffusion speed $\alpha = 50$ and simulates using an adaptive second order Runge–Kutta solver with third order error correction Virtanen et al. (2020) up to time $T = 2000$.
- Lloyd's algorithm and IIDMS use 1000 steps.

Now, we describe the details of Section 4.1.

- As a benchmark for optimal quantization with M points, we use the square root of the $(M+1)$-st eigenvalue of the integration operator associated with the kernel κ and the probability measure π, divided by $\|\boldsymbol{v}_0\|_{\mathcal{H}}$. The eigenvalues of this operator are classically used in measuring the complexity of numerical integration in an RKHS; see Pinkus (2012); Bach (2017).
- All experiments were performed in Python using Numba (Lam et al., 2015).
- Computationally, we use a parallelized CPU implementation of the algorithms. These were run on a cluster using up to 32 tasks per simulation. Many of the smaller visualizations were produced from experiments on a local machine with 16 CPU cores and 32 GB.
- For WFR, we use a fixed step-size Runge–Kutta fourth order ODE solver for simulations. As this requires four evaluations of $\boldsymbol{v}_0$ and $\bar{\boldsymbol{v}}_1$ for each iteration, we only run the ODE for one fourth as many iterations as the other algorithms. Outside of this experiment setting, we suggest that common *adaptive* ODE solvers should provide better results. We also choose the diffusion rate α according to naïve hyperparameter tuning, where simple experiments provide good results for about $\alpha = 25$.
- For MSIP, we use the damping factor $\eta = 0.8$ for the fixed-point iterations. Moreover, we use standard floating-point methods to ensure stability when working with terms that take extremely small values such as the functions $\boldsymbol{v}_0$ and $\bar{\boldsymbol{v}}_1$, which we employ for the squared exponential kernel.
- For MMDGF, we use a step-size of 10^{-1} and inject noise at each iteration via $\varepsilon_t \sim \mathcal{N}(0, \beta/\sqrt{t})$, where t is the iteration and $\beta = 0.05$. This noise injection is recommended in Arbel et al. (2019) to improve convergence rates and guarantees.
- For IFTFlow, we chose to implement the algorithm described in Gladin et al. (2024, Appendix A) with a regularization on the weight parameter of 10^{-3} and, similar to Gladin et al. (2024), we perform the MMD geometry minimization via a single step of projected gradient descent.
- We observe that the methods proposed in both Arbel et al. (2019) and Gladin et al. (2024) draw new samples from the target distribution at every iteration t (and hence assume access to an infinite number of samples from the target distribution); here, however, we assume a fixed target distribution in each simulation and thus assume that the same N samples are available across all iterations of each algorithm.

Finally, we note that, for the MNIST experiment, we use a Matérn kernel with a bandwidth of $\sigma = 2.25$ and smoothness of $\nu = 1.5$.

A.3 Target distributions in Section 4.1

We consider the following setting: π is the empirical measure associated with $N = 1000$ i.i.d. samples from a mixture of Gaussians $\sum_{m=1}^{L_0} \pi_m \mathcal{N}(c_m, \Sigma_m)$, where $L_0 \in \mathbb{N}^*$ is the number of components of the mixture, $c_1, \dots, c_{L_0} \in \mathbb{R}^d$ are the means of the mixtures, and $\Sigma_1, \dots, \Sigma_{L_0} \in \mathbb{S}_d^{++}$ are the covariance matrices, and $\pi_1, \dots, \pi_{L_0}$ are the positive weights of the mixture which satisfy $\sum_{m=1}^{L_0} \pi_m = 1$. We consider two versions of this mixture distribution: one in dimension $d = 2$ with $L_0 = 3$, and another in dimension $d = 100$ with $L_0 = 5$. Figure 4 illustrates some marginals of the 100-dimensional distribution used in Section 4.1. The 100-dimensional distribution is marginally normalized to have a covariance matrix with ones on the diagonal.

A.4 Visualization of paths

We provide a qualitative comparison of the transient behavior of the following algorithms: i) Wasserstein–Fisher–Rao Flow interacting particle system for MMD (WFR-IPS) introduced in Section 3.1, ii) mean shift interacting particles (MSIP) introduced in Section 3.2, iii) IFTflow introduced in Gladin et al. (2024), iv) MMD gradient flow (MMDGF) introduced in Arbel et al. (2019). Figure 5 compares the dynamics of the four algorithms for their first 1000 iterations. For each algorithm, we illustrate the dynamics of matching $M = 3$ nodes to a mixture of three Gaussians. For WFR-IPS and MSIP, the left and right plots in Figure 5(a) and (b) correspond to two different initializations: random elements of the dataset (left), and points far from the support of π

(right). For each of IFTFlow and MMDGF, we demonstrate both initializations in the same plot. The lines indicate particle trajectories, where overlapping markers indicate slower dynamics. In particular, we see WFR-IPS performing well in both scenarios with sufficient iterations. MSIP is significantly faster; though it produces more erratic trajectories than WFR and might first overshoot the support, it converges at the end. In contrast, both MMDGF and IFTFlow succeed when starting from the support of the target, but perform poorly otherwise.

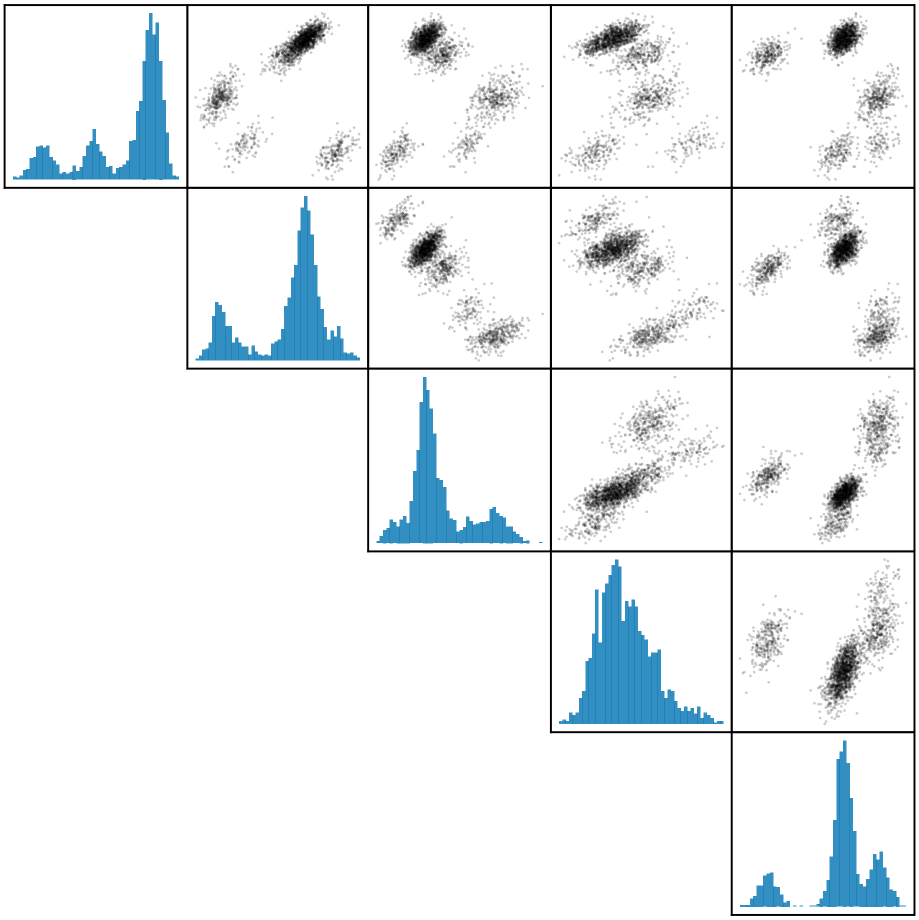
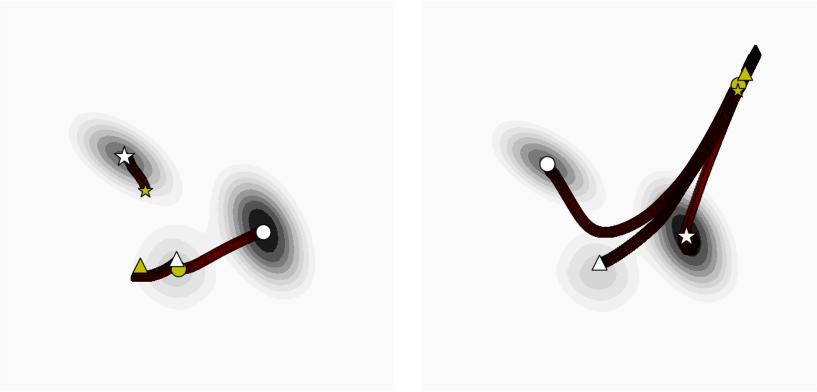
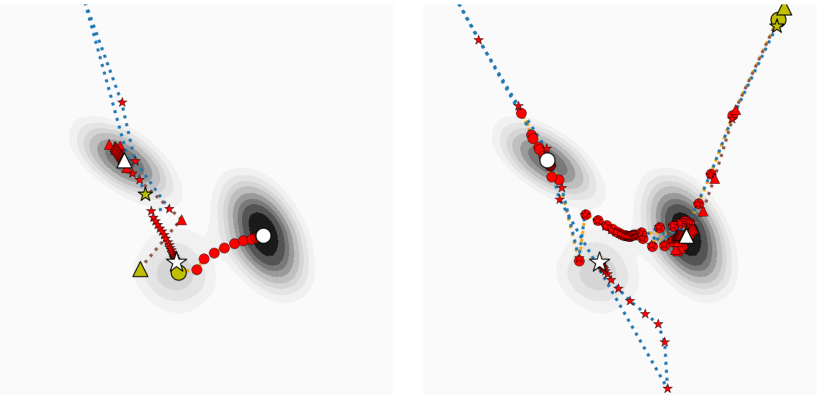


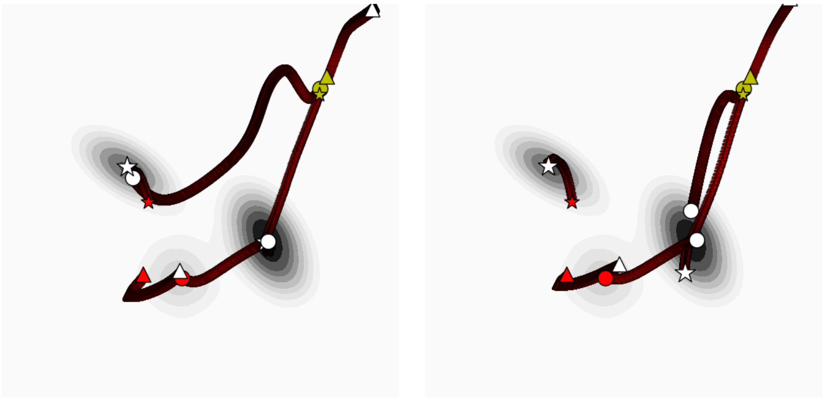
Figure 4: First five univariate and pairwise marginals of the 100-dimensional distribution used in Section 4.1



(a) WFR-IPS



(b) MSIP



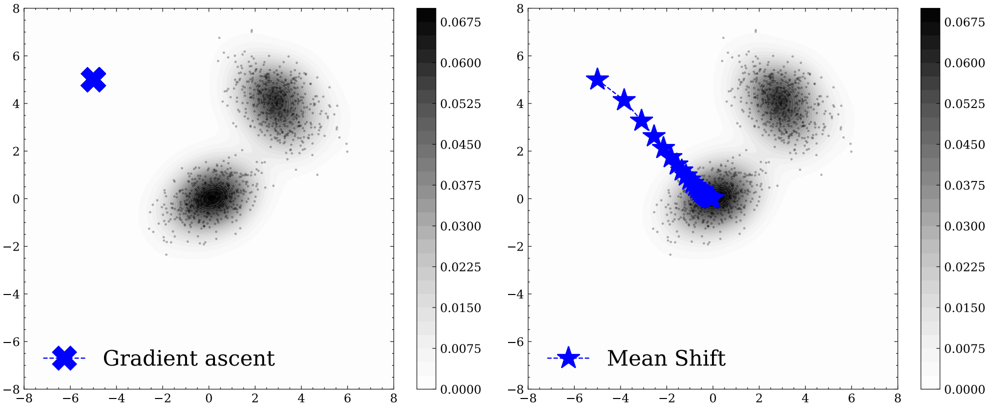
(c) IFTFlow

(d) MMDGF

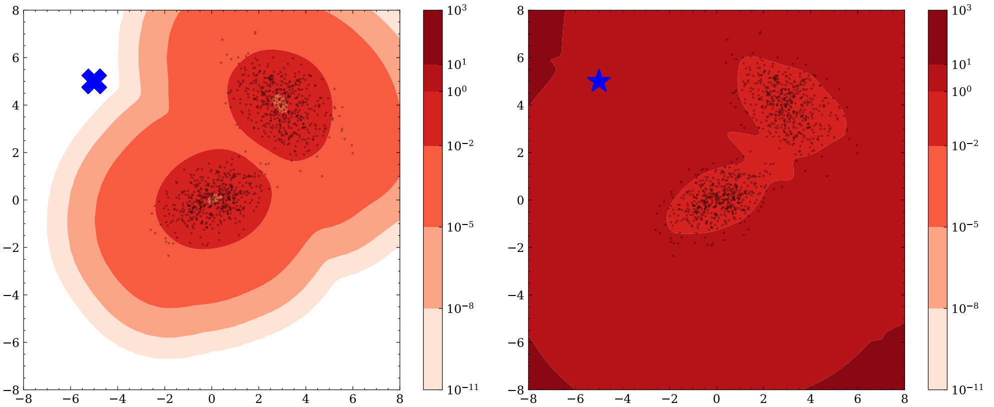
Figure 5: Trajectories of four algorithms started at two different intializations (yellow and red symbols). Each marker is one iteration for a particle, and the lines show particle paths. White markers are the final particle positions.

A.5 Mean shift vs. MMD gradient flow for $M = 1$

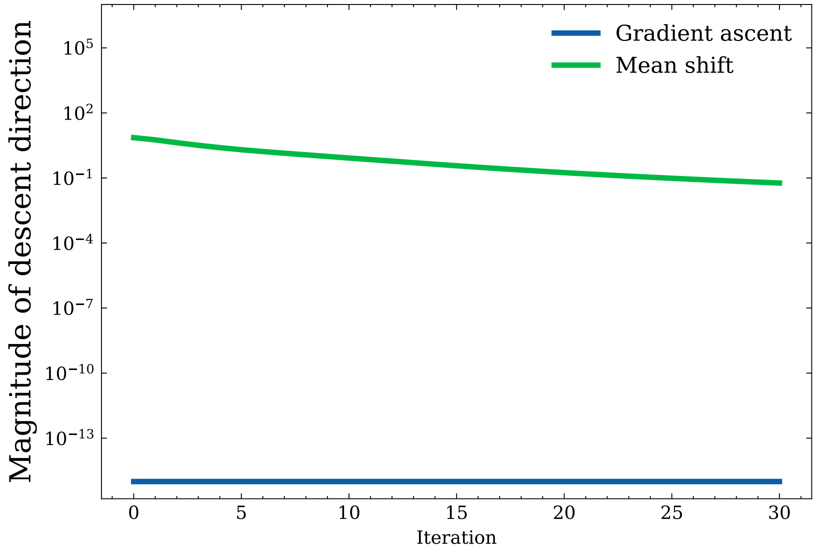
Figure 6a compares mean shift with the algorithm derived in Arbel et al. (2019) for $M = 1$, which we will refer to as gradient ascent on the KDE v_0. The target π is an N-sample empirical representation of a mixture distribution on $\mathbb{R}^2$. The two algorithms are initialized from the same point. We observe that mean shift converges to one of the two modes within a few iterations, whereas the particle performing gradient ascent on the KDE remains nearly stationary across iterations. The difference between the two algorithms can be explained by comparing the vector field ∇v_0 to the vector field $\nabla v_0 / v_0$. The former governs the dynamics of gradient ascent; the latter governs the



(a) Dynamics of the algorithms: (left) gradient ascent on the KDE v_0, corresponding to MMDGF wtih $M = 1$; (right) mean shift.



(b) Norms of the descent direction vector fields of the two algorithms. Initialization for each algorithm marked in blue. Colorbars and contour levels are identical in the left and right plots. (Left): $\|\nabla v_0\|$. (Right): $\|\nabla v_0\|/v_0$).



(c) Comparison of the norm of the descent directions of the two algorithms across iterations.

Figure 6: Comparison of the dynamics of mean shift and the discretization of MMD gradient flow proposed in Arbel et al. (2019).

dynamics of mean shift. Figure 6b compares the norms of the vector fields associated with the two algorithms. We observe that $\|\nabla v_0(y)\|$ becomes extremely small when y lies far from the support of the distribution, unlike $\|\nabla v_0(y)/v_0(y)\|$, which takes significant values even when y is far from the support. This observation is further supported by Figure 6c, which shows the evolution of the vector field norms across iterations for both algorithms.

A.6 Additional results for the MNIST simulation

A.6.1 The effect of additional iterations

Figure 7 shows the results of the simulation detailed in Section 4.2 after 15000 iterations (10000 more iterations than in Figure 3) for MSIP, WFR-IPS, Lloyd's algorithm, and IFTFlow. We observe that their recovered images are less blurry compared to the results after 5000 iterations. Interestingly, IFTFlow ends with one particle on the target's support despite not having any such particles in Figure 3; this particle seems to correspond visually with the mode that IIDMS finds in Figure 3.

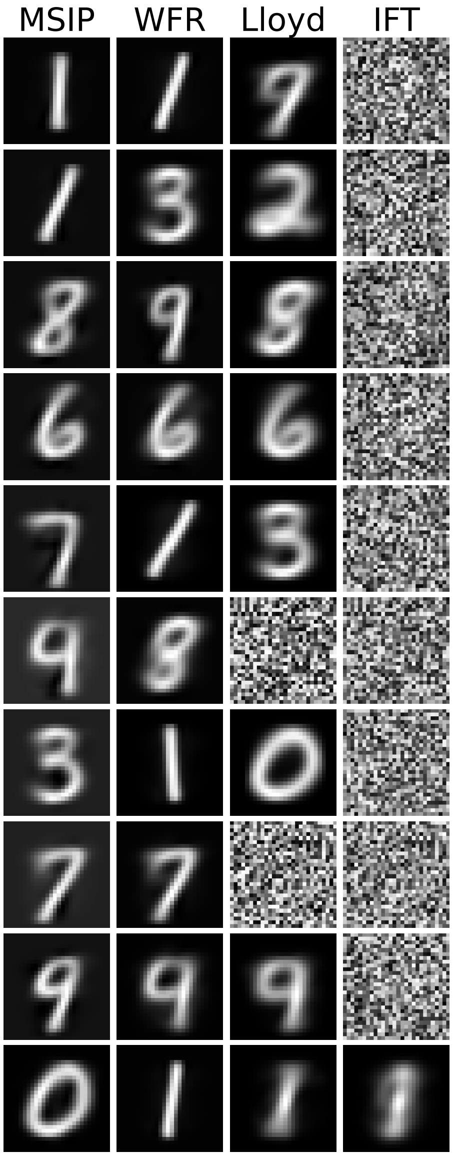


Figure 7: Comparison of four algorithms on MNIST for the iteration $T = 15000$.

A.6.2 Quantitative results for MNIST

In Figure 8, we evaluate the quantitative performance of each algorithm's quantization of MNIST. Specifically, we calculate the MMD for each algorithm using the Matérn kernel with $\sigma = 2.25$ and $\nu = 1.5$. We use 10 different initializations for the experiment and show the median MMD (over these initializations) marginally at each iteration. Despite the fact that DMGD starts from a relatively low value of MMD, this value does not decrease over time for this algorithm. MSIP and WFR-IPS are the only two algorithms that enjoy substantial dynamics in these simulations, shown by their reduction of MMD over time.

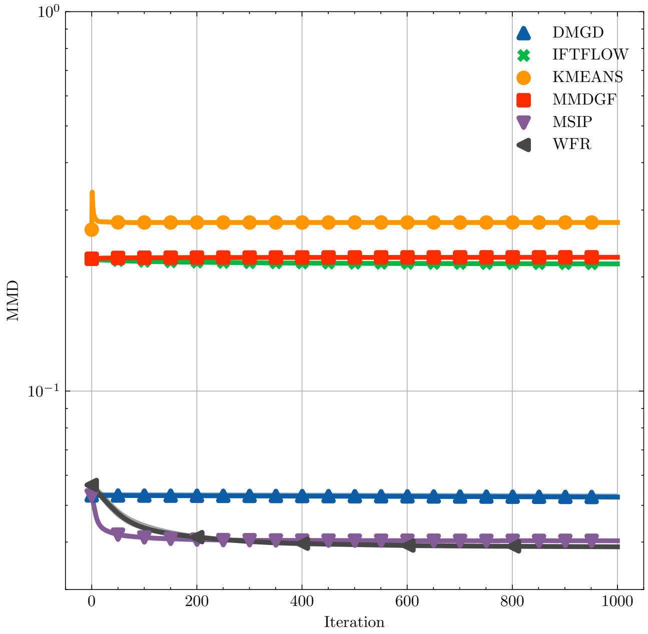


Figure 8: Comparison of different algorithms' quantization of MNIST with $M = 10$ points, up to iteration $T = 1000$.

A.7 Comparison to kernel thinning

Kernel thinning, a recent technique in coreset construction Dwivedi and Mackey (2024), is another kernel-based approach to summarizing datasets. Some brief comparisons and contrasts to our approaches are as follows:

- Kernel thinning methods are restricted to selecting points that are in the support of the data, whereas the methods discussed here optimize point positions over the (continuous) space $\mathcal{X}^M$;
- As such, kernel thinnings are finite-time calculations, with complexity often on the order of $\mathcal{O}(N^2)$ Dwivedi and Mackey (2024, 2022). Neglecting M, the methods discussed in our work have a complexity $\mathcal{O}(TN)$ for T iterations; whether these algorithms are faster than kernel thinning then depends on their convergence rates, e.g., if T is smaller than N (which is often the case when N is very large), and on how the cost of each iteration scales with M.
- The number of summarization points M in kernel thinning is restricted to values $M = \mathrm{floor}(N/2^r)$ for some integer r due to "halving rounds," whereas there are no such restrictions on the choice of M for our proposed methods;
- Unlike flow-based methods, kernel thinning does not involve a randomly chosen initial condition. Instead, stochasticity is introduced through splitting and swapping within the coreset;
- The standard kernel thinning discussed in Dwivedi and Mackey (2024) is *unweighted*. There are algorithmic extensions to weighted coresets in this line of literature, but the construction of these weighted coresets often rely on evaluating the target score $\nabla \log \pi$.

Table 2: Kernel thinning error quantification. M is the number of nodes in our methods and M_{kt} is the size of permissible coreset for kernel thinning. All error metrics are reported via the interval between the middle two quartiles (25th and 75th percentiles) of performance over 32 runs.

	Our methods			Kernel thinning		
Example	M	WFR-IPS	MSIP	M_{kt}	MMD_u	MMD_w
Joker	10	[0.164, 0.174]	[0.156, 0.159]	9	[0.171, 0.181]	[0.149, 0.158]
GMM-2	3	[0.0812, 0.144]	[0.0812, 0.0812]	3	[0.343, 0.346]	[0.167, 0.171]
GMM-100	10	[0.0493, 0.119]	[0.0282, 0.0438]	7	[0.342, 0.344]	[0.111, 0.112]
				15	[0.228, 0.229]	[0.0992, 0.101]

While the method in Dwivedi and Mackey (2024) yields unweighted quantization, we also investigate a weighted variant using (4). Thus, we quantify the MMD of kernel thinning (KT) on a few examples in two ways: 1) the MMD on the unweighted coreset produced by KT using the same kernel we use to quantify error in Section 4 (MMD_u), 2) the MMD using *optimal* weights (4) for result of KT (MMD_w). For each example, we swap and split in KT via the same kernel we use for MSIP and WFR-IPS.

We compare results of KT to MSIP and WFR-IPS in Table 2, where we report our results from Figure 2 for the GMM examples, and also show the results from each algorithm on one hundred random initializations from the joker distribution of Figure 1. For the latter distribution, we use a step size of 0.3 for MSIP with 5000 steps, which is a very conservative choice to ensure that we avoid singularities. We simulate up to time 2000 with WFR-IPS, where $\alpha = 10$, with a Runge–Kutta second order solver with third order adaptive error correction. We do not show the performance of KT on the MNIST dataset, as the time complexity of kernel thinning for such a large dataset makes the comparison unreasonable.

A.8 Additional example configurations

In Figure 9, we show the weighted quantizations obtained using MSIP and WFR-IPS, alongside weighted quantizations from Lloyd's algorithm, for a few two-dimensional distributions with $M = 5$ and $M = 15$. The distributions are an asymmetric "rings" distribution (each ring j has a density in polar coordinates (r, θ)

given by $\pi_j(r,\theta) = U(\theta; 0, 2\pi)\mathcal{N}(r; 1, 0.05^2))$, a "checkers" distribution (i.e., a mixture of uniform densities $\pi(x) = \frac{1}{L_0}\sum_{j=1}^{L_0} U(x - [z_j^{(1)}, z_j^{(2)}]; [0,1]^2))$, and Neal's two-dimensional funnel distribution Neal (2003) (i.e., $\pi(x_1, x_2) = \mathcal{N}(x_1; 0, 1)\mathcal{N}(x_2; 0, \exp(x_1)))$.

We see that, when the number of particles is small in comparison to number of modes of the distribution, it becomes difficult to maintain nodes on the support of the distribution, particularly when the distribution has discontinuous density. This is easily explained by noting that even when the underlying π is discontinuous, v_0 (its KDE) is continuous with support everywhere. Regardless, MSIP and WFR-IPS do a better job of minimizing the MMD than Lloyd's algorithm, as seen in Table 3. We also note that MSIP and WFR-IPS achieve similar final quantization performance in most cases (with the exception of the funnel distribution for $M = 5$); again, this is to be expected, since both algorithms seek to minimize the MMD, and MSIP further seeks a stationary point of the WFR flow.

Table 3: MMD values for configurations shown in Figure 9.

	Rings		Checkers		Funnel	
	$M = 5$	$M = 15$	$M = 5$	$M = 15$	$M = 5$	$M = 15$
Bandwidth	0.8	0.4	1.0	0.4	4.0	0.8
MSIP	**0.132**	**0.0815**	0.0947	**0.0680**	0.113	0.0466
WFR	**0.132**	0.0830	**0.0930**	0.0682	**0.0257**	**0.0441**
Lloyd's	0.203	0.122	0.143	0.116	0.140	0.184

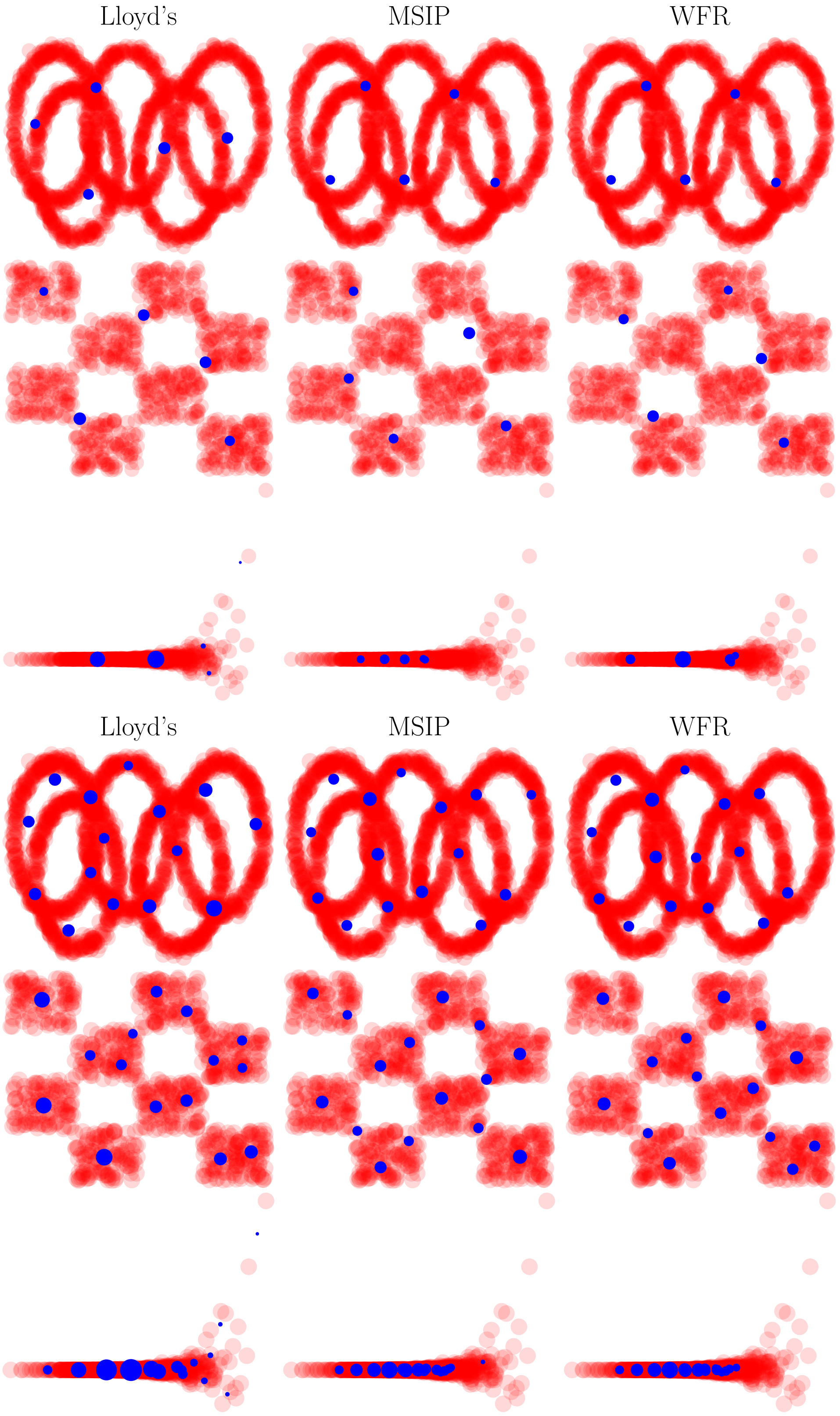


Figure 9: Final configuration of MSIP and WFR-IPS compared to Lloyd's algorithm with identical initialization for $M = 5$ (top) and $M = 15$ (bottom). Particle size reflects quantization weight.

A.9 Weight positivity

We see, using logarithmic horizontal scaling in Figure 10, that WFR-IPS in practice produces weights in $(0,1)^d$ across twenty runs for the checkers and rings distributions. These two-dimensional examples are run with a Runge–Kutta second order integrator with third-order error correction. Similarly, we examine the MSIP configurations at the *final* time in Figure 11. We observe **positivity of the final weights across all one hundred random initializations**. Over all one hundred *entire* trajectories for all $M = 5$ weights, only 0.0628% of weights are nonpositive in the checkers example and 0.0472% in the rings example. These results indicate that, while negative weights may happen in MSIP iterations, they are not observed near the steady state, nor are they common in the transient state.

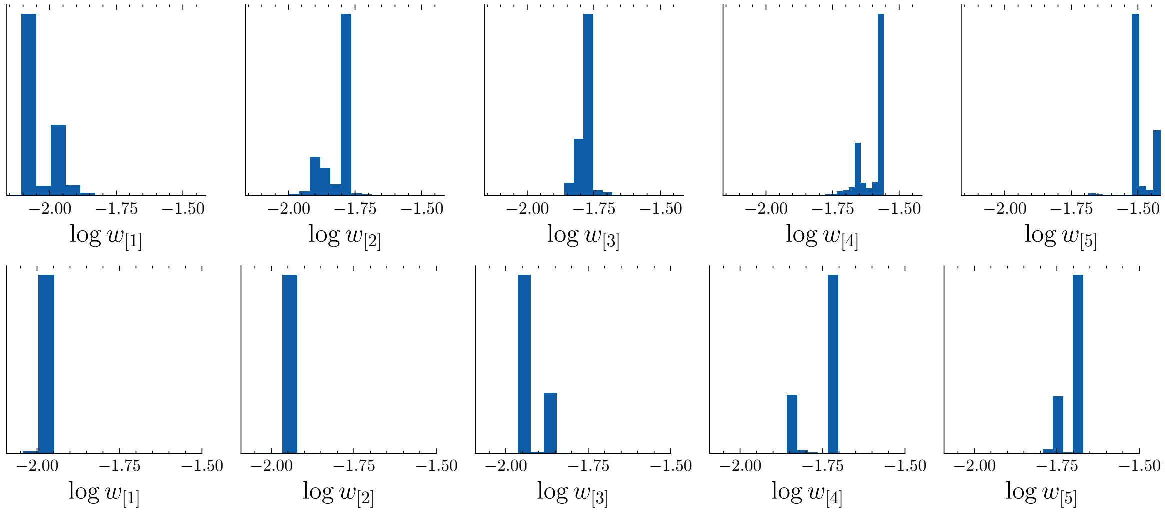


Figure 10: Weights of WFR-IPS trajectories, marginalizing out time: The weights are sorted at each timestep, and the distribution across time is plotted above, where weights increase left to right (ordering statistic subscript $[j]$ is the jth smallest). (Top): Checkers target. (Bottom): Rings target.

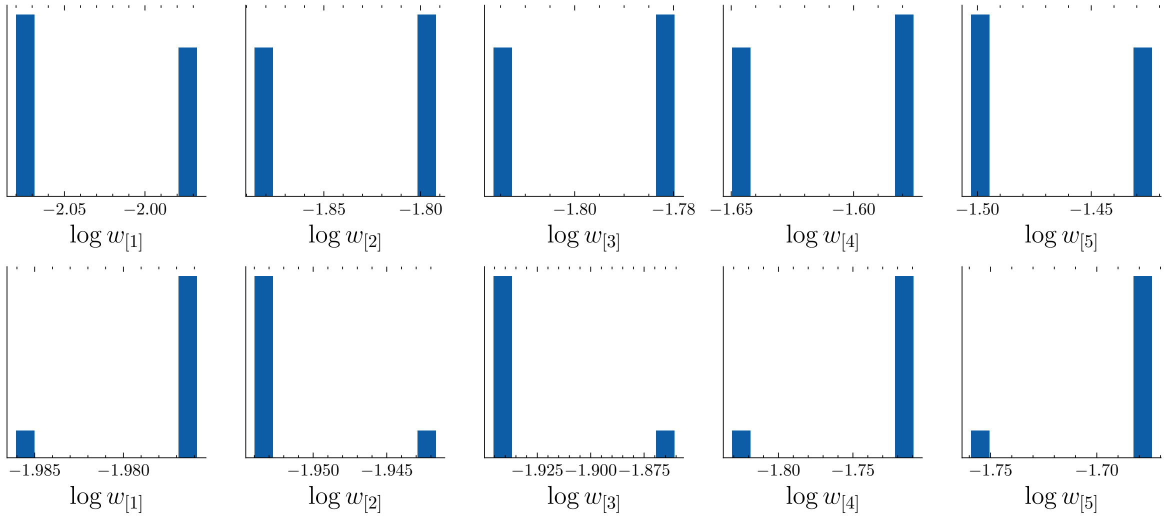


Figure 11: Weights of MSIP *final* configurations. The weights increase from left to right (ordering statistic subscript $[j]$ is the jth smallest). (Top): Checkers target. (Bottom): Rings target.

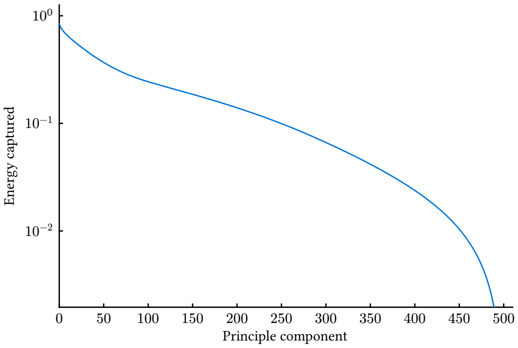


Figure 12: Residual squared energy from first 500 principle components. Given the left singular vectors u_i and singular values σ_i of the centered samples, the curve f is defined as $f_i = \frac{\sum_{j=i+1}^{500} \sigma_j^2}{\sum_{j=1}^{500} \sigma_j^2}$.

B Application to cell photography dataset

We use a dataset furnished by Acevedo et al. (2019), which consists of photographs for seven different types of cells in a bloodstream, where each photograph is 360×363 pixels (i.e. a true dimension of 131040 per sample for each color channel). We then convert these photos to grayscale and reduce the dimensionality of the photographs using principle component analysis (PCA). While this is not sufficient for reproducing the data (the algebraic decay of the first 500 singular values can be seen in Figure 12), it is adequate for clustering, enabling us to distinguish between cell types and capture intra-class variability. In Figure 13, we visualize the different cell groupings when projected onto the first five principle components—note that there is natural groupings of the images, which may be overlapping when viewing low-dimensional projections. In this example, we keep the first 100 principle components then run Lloyd's algorithm and MSIP each for 500 iterations. We initialize each with 21 samples from the dataset, where MSIP uses a squared-exponential kernel with bandwidth 0.025.

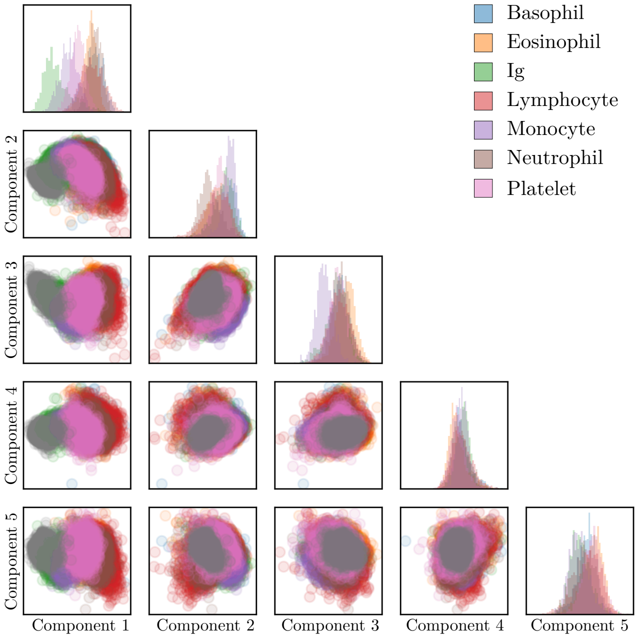


Figure 13: First five principle components of the dataset in Section B, colored by the true label of the sample.

We note that both Lloyd's algorithm and MSIP finish 500 iterations on this $d = 100$ dimensional dataset over 16339 samples for 21 centroids in less than ten seconds of wall-clock time.[6] This clearly shows practicality in the

---

[6] This was tested on an Apple 12-core M2 Pro processor using the provided code.

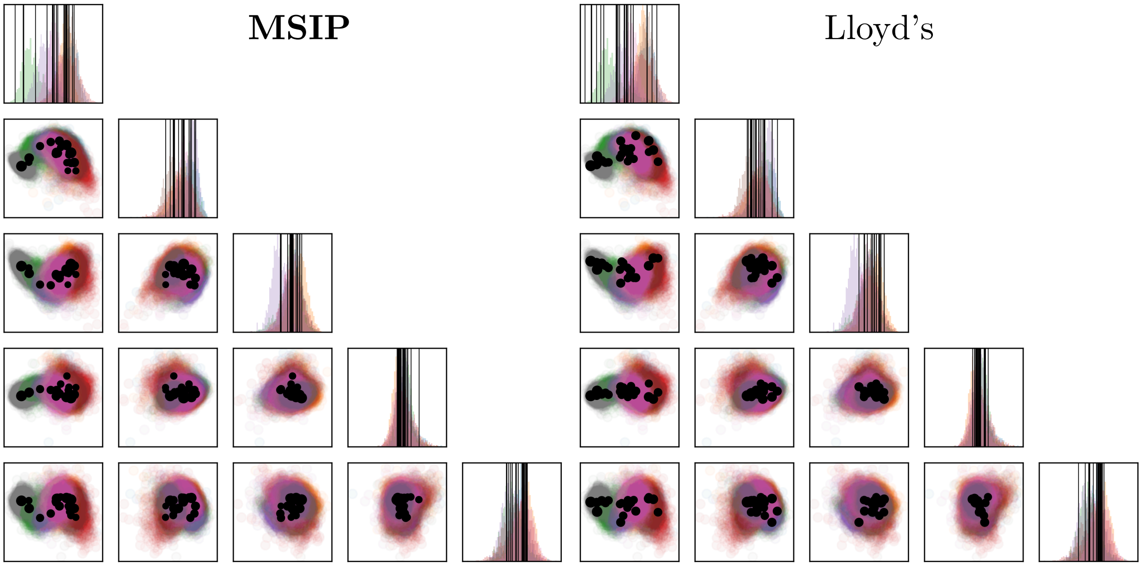


Figure 14: Visualizing the centroids obtained for Section B given a fixed initialization. Left: MSIP. Right: Lloyd's. We overlay the particle marginal locations onto the marginal histograms as vertical lines, and use black points (whose radius corresponds to the weight of the particle) for the two-dimensional marginals. The axes and remaining colors are identical to those in Figure 13.

low-centroid regime. Further, this can be accelerated greatly by more efficient implementations in the future.

Figure 15 shows the evolution of the MMD for the two algorithms over the iterations. We observe that MSIP outperforms Lloyd's regardless of the initialization, which is expected by the construction of the method. Figure 14 shows the dataset projected onto its first two principal directions, along with the projections of the particles obtained by both algorithms. We observe that both algorithms yield configurations covering the distribution in some sense. Now, we compare the intra-centroid distances for each of the two algorithms to gauge diversity of the particles. MSIP achieves a minimum intra-centroid distance of 0.0295 for the example in Figure 14, while Lloyd's algorithm only 0.0249. On the other hand, the largest intra-centroid distance for MSIP is 0.0439, compared to 0.0422 for Lloyd's algorithm. In other words, the configuration obtained by MSIP has higher variance while capturing the distribution better (in the MMD).

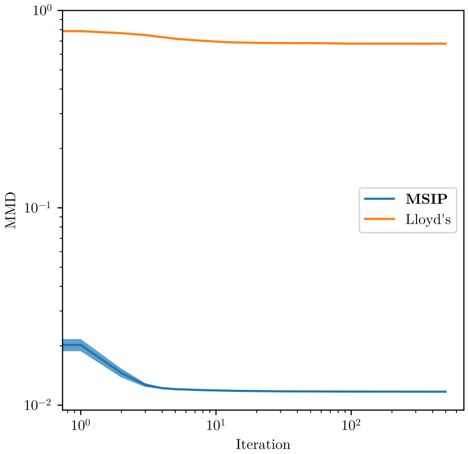


Figure 15: Convergence of the MMD for each of Lloyd's and MSIP over ten independent trials, where each algorithm is initialized identically to the other for each trial. MMD measured with bandwidth 0.025.

C Existing work on gradient flows for MMD minimization

In the following, we review a few constructions of gradient flows for different choices of metric D to minimize the MMD functional

$$\mathcal{F}[\mu] := \frac{1}{2} \operatorname{MMD}(\mu, \pi)^2. \tag{34}$$

When the metric D corresponds to the MMD,[7] the equation (14) simplifies to $\dot{\mu}_t = \mu_t - \pi$, whose solution is given by $\mu_t = e^{-t}\mu_0 + (1 - e^{-t})\pi$, where μ_0 is the initial condition (Gladin et al., 2024, Theorem 3.4). This solution is not suitable for the quantization problem we consider, as π generally cannot be expressed as a mixture of M Diracs.

Arbel et al. (2019) studied the gradient flow of $\mathcal{F}$ under the 2-Wasserstein (W_2) geometry, corresponding to the solution of

$$\dot{\mu}_t = \operatorname{div}\left(\mu_t \nabla \mathcal{F}_\delta[\mu_t]\right), \tag{35}$$

where $\mathcal{F}_\delta[\mu]$ is the first variation of $\mathcal{F}$ at μ given by

$$\mathcal{F}_\delta[\mu](\cdot) = \int_{\mathcal{X}} \kappa(\cdot, x)\, \mathrm{d}\mu(x) - \int_{\mathcal{X}} \kappa(\cdot, x)\, \mathrm{d}\pi(x). \tag{36}$$

The authors propose various numerical solutions to (35) based on discretizations in time and space. In particular, they replace the measures μ_t by empirical counterparts, yielding a system of interacting particles. Moreover, they derive conditions for the convergence of their numerical schemes, and study the importance of *noise injection* as a regularization method. However, the approaches deteriorate dramatically when initialized poorly, requiring a large number of particles to converge. Further, rigidly constraining particles to have uniform weights is problematic; this can be seen for $M = 1$ when comparing gradient descent on the MMD to mean shift, which we discuss in Section 3.2.

Altekrüger et al. (2023) and Hertrich et al. (2024) also study Wasserstein gradient flow of MMD, exploiting the particular structure induced by Riesz kernels to develop more efficient algorithms in the context of neural sampling schemes.

An alternative geometry called *spherical interaction-force transport* (IFT) is proposed and studied in Gladin et al. (2024); Zhu and Mielke (2024). Inheriting properties of both the W_2 and MMD geometries, the IFT gradient flow satisfies

$$\dot{\mu}_t = \alpha \operatorname{div}\left(\mu_t \nabla \mathcal{F}_\delta[\mu_t]\right) - \beta(\mu_t - \pi), \tag{37}$$

where $\alpha, \beta \in \mathbb{R}$ are positive. The authors prove that if μ_t solves (37), then $\mathcal{F}[\mu_t]$ decays exponentially with t (Gladin et al., 2024, Theorem 3.5). While this is a remarkable theoretical result, it only holds in the mean-field limit $M \to +\infty$. They discretize (37) using a system of weighted interacting particles; as we show in Section A.4, many of these particles tend to drift far away from the modes of π, making the resulting quantization inefficient for, e.g., quadrature. Further, the proposed implementation is complicated by an inner M-dimensional constrained optimization problem at every iteration.

Finally, the Wasserstein–Fisher–Rao (WFR) geometry allows both mass transport and total mass variation. The PDE corresponding to the WFR metric is

$$\dot{\mu}_t = \alpha \operatorname{div}(\mu_t \nabla \mathcal{F}_\delta[\mu_t]) - \beta\, \mathcal{F}_\delta[\mu_t]\mu_t. \tag{38}$$

This corresponds to a gradient flow of the functional $\mathcal{F}$, extending the W_2 distance to measures with different total masses, i.e., *unbalanced* optimal transport. See (Kondratyev et al., 2016; Gallouët and Monsaingeon, 2017; Chizat et al., 2018; Liero et al., 2018) for details about the WFR metric, its geometric properties, and the construction of WFR gradient flows. Additionally, there are theoretical and numerical results for the minimization of a kernelized KL divergence as the energy (compared to our usage of the MMD) using the WFR geometry (Lu et al., 2019, 2023; Yan et al., 2024).

Compared to (35), Equation (16) introduces $\mathcal{F}_\delta[\mu_t]\mu_t$ as a reaction term, which allows the total mass of μ_t to change as a function of t. Unbalanced transport allows for this creation or destruction of mass, which might

---

[7] Again, not to be confused with when $\mathcal{F}$ is the MMD.

be helpful when mass conservation is not a natural assumption, as in the case of the minimization problem (1). Indeed, the weights in this optimization problem do not necessarily lie on the simplex, so the resulting quantization is not, in general, a probability measure. Despite this formulation, the numerical scheme proposed by Gladin et al. (2024) to solve (16) artificially conserves mass by projecting the weights onto the simplex; this is inconsistent with the PDE's theoretical properties.

D Proofs

D.1 Proof of Proposition 3.1

This is a ready extension to common proof techniques in, e.g., (Yan et al., 2024) and (Chewi et al., 2024, Section 5.7.). Let $\varphi \in C_c^\infty$ be a test function. We have

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbb{R}^d}\varphi(x)\mu_t(\mathrm{d}x) &= \frac{\mathrm{d}}{\mathrm{d}t}\left[\sum_{i=1}^M w_i^{(t)}\varphi(y_i^{(t)})\right]\\ &= \sum_{i=1}^M \frac{\mathrm{d}}{\mathrm{d}t}w_i^{(t)}\varphi(y_i^{(t)}) + \sum_{i=1}^M w_i^{(t)}\frac{\mathrm{d}}{\mathrm{d}t}\varphi(y_i^{(t)})\\ &= \sum_{i=1}^M \frac{\mathrm{d}}{\mathrm{d}t}w_i^{(t)}\varphi(y_i^{(t)}) + \sum_{i=1}^M w_i^{(t)}\langle\nabla\varphi(y_i^{(t)}), \frac{\mathrm{d}}{\mathrm{d}t}y_i^{(t)}\rangle\\ &= -\sum_{i=1}^M \mathcal{F}_\delta[\mu_t](y_i^{(t)})w_i^{(t)}\varphi(y_i^{(t)}) - \alpha\sum_{i=1}^M w_i^{(t)}\left\langle\nabla\varphi(y_i^{(t)}), \nabla\mathcal{F}_\delta[\mu_t](y_i^{(t)})\right\rangle\\ &= -\int_{\mathbb{R}^d}\mathcal{F}_\delta[\mu_t](x)\varphi(x)\mu_t(\mathrm{d}x) - \alpha\int_{\mathbb{R}^d}\langle\nabla\varphi(x), \nabla\mathcal{F}_\delta[\mu_t](x)\rangle\,\mu_t(\mathrm{d}x)\\ &= -\int_{\mathbb{R}^d}\varphi(x)\mathcal{F}_\delta[\mu_t](x)\mu_t(\mathrm{d}x) + \alpha\int_{\mathbb{R}^d}\varphi(x)\mathrm{div}(\mu_t\nabla\mathcal{F}_\delta[\mu_t])(x)\mathrm{d}x.\end{aligned}$$

In other words, $(\mu_t)_{t\geq 0}$ satisfies (16) in the sense of distributions.

D.2 Proof of Proposition 3.2

Preliminary bounds Since $x \mapsto \kappa(x,x)$ is a constant $\kappa(x,x) \equiv B_\kappa$ for all $x \in \mathcal{X}$, we have

$$\begin{aligned}\forall x,y\in\mathcal{X},\ \ |\kappa(x,y)| &= |\langle\kappa(x,.),\kappa(y,.)\rangle_{\mathcal{H}}|\\ &\leq \|\kappa(x,.)\|_{\mathcal{H}}\|\kappa(y,.)\|_{\mathcal{H}}\\ &= \sqrt{\kappa(x,x)\kappa(y,y)} = B_\kappa.\end{aligned} \tag{39}$$

In particular, we have

$$\forall y\in\mathcal{X},\ \ |v_0(y)| \leq \left|\int_{\mathcal{X}}\kappa(x,y)\mathrm{d}\pi(x)\right| \leq \int_{\mathcal{X}}|\kappa(x,y)|\,\mathrm{d}\pi(x) \leq B_\kappa\int_{\mathcal{X}}\mathrm{d}\pi(x) = B_\kappa. \tag{40}$$

Moreover, denote by B'_κ an upper bound on the gradient of the kernel κ. We have $\nabla v_0(y) = \int_{\mathcal{X}}\nabla_2\kappa(x,y)\mathrm{d}\pi(x)$ for any $y \in \mathcal{X}$, so that

$$\|\nabla v_0(y)\| \leq \int_{\mathcal{X}}\|\nabla_2\kappa(x,y)\|\mathrm{d}\pi(x) \leq \int_{\mathcal{X}}B'_\kappa\mathrm{d}\pi(x) = B'_\kappa. \tag{41}$$

Step 1: Existence The system of coupled ODEs (17) can be written as $\dot{z} = F(z)$, where $z \in \mathbb{R}^{M(d+1)}$ defined by the concatenation of $y_1,\ldots,y_M,\boldsymbol{w}$ into an $M(d+1)-$dimensional vector,

$$z^\top = (y_1^\top,\ldots,y_M^\top,w_1,\ldots,w_M), \tag{42}$$

and $F:\mathbb{R}^{M(d+1)}\to\mathbb{R}^{M(d+1)}$ is defined as $F(z)^\top = (F_{y,1}(z)^\top,\ldots,F_{y,M}(z)^\top,F_w(z)^\top)$, where

$$\forall i\in[M],\ \ F_{y,i}(z) := -\alpha\left(\sum_{m=1}^M w_m\nabla_2\kappa(y_m,y_i) - \nabla v_0(y_m)\right), \tag{43}$$

and

$$\forall i \in [M],\;\; F_w(z)_i := -w_i\left(\sum_{m=1}^{M} w_m \kappa(y_m, y_i) - v_0(y_m)\right). \tag{44}$$

By assumption, κ and $\nabla\kappa$ are continuous. Therefore, v_0 and ∇v_0 are continuous and F is continuous over $\mathbb{R}^{M(d+1)}$.

By the Peano existence theorem (Teschl, 2012, Theorem 2.19), for any $z_0 \in \mathbb{R}^{M(d+1)}$, the initial value problem

$$\begin{cases} \dot{z} & = F(z) \\ z(0) & = z_0 \end{cases} \tag{45}$$

has at least one solution $\hat{z} : I \to \mathbb{R}^{M(d+1)}$ defined on a neighborhood I of 0 that satisfies $\hat{z}(0) = z_0$. In the following, denote $\tilde{t}$ as a positive scalar such that $[0, \tilde{t}\,] \subset I$.

Step 2: Positivity of the weights and universal boundedness of the weights Since $\hat{z}$ is continuous on $[0, \tilde{t}]$, it is bounded. In the following, we denote by $B_{\tilde{t}} > 0$ that bound. We prove in the following that the corresponding weights $w_1^{(t)}, \dots, w_M^{(t)}$ are positive when $t \in [0, \tilde{t}\,]$.

For that, let $i \in [M]$, and assume that there exists $t \in (0, \tilde{t}]$ such that $w_i^{(t)} \leq 0$. Denote $t^* = \min\big\{t \in [0, \tilde{t}] | w_i^{(t)} \leq 0\big\}$. By definition, $\forall t \in [0, t^*]$, $w_i^{(t)} > 0$. Moreover, observe that for $i \in [M]$ and $t \in [0, \tilde{t}\,]$ we have

$$\left|\sum_{m=1}^{M} w_i^{(t)} \kappa(y_m^{(t)}, y_i^{(t)}) - v_0(y_i^{(t)})\right| \leq \sum_{m=1}^{M} |w_i^{(t)}|\, \kappa(y_m^{(t)}, y_i^{(t)}) + |v_0(y_i^{(t)})| \leq (B_{\tilde{t}} M + 1) B_\kappa. \tag{46}$$

Therefore $\forall t \in [0, t^*]$, $\dot{w}_i^{(t)} \geq -\big((B_{\tilde{t}} M + 1) B_\kappa + 1\big) w_i^{(t)}$. Thus, using Grönwall's lemma, we prove that $w_i^{(t)} \geq w_i^{(0)} e^{-\left((B_{\tilde{t}} M+1)B_\kappa+1\right)t}$ for all $t \in [0, t^*]$, and as a result $w_i^{(t^*)} > 0$, which is a contradiction. Therefore, for any $t \in [0, \tilde{t}\,]$, we have $w_i^{(t)} > 0$.

Now, we prove that for any $i \in [M]$, we have

$$\forall t \in [0, \tilde{t}\,],\;\; w_i^{(t)} < 1. \tag{47}$$

For that, observe that by the non-negativity of the kernel ($\kappa(y_1, y_2) \geq 0$ for any $y_1, y_2 \in \mathcal{X}$) and the weights on $[0, \tilde{t}\,]$, we have

$$\forall t \in [0, \tilde{t}\,],\;\; \sum_{m=1}^{M} w_m^{(t)} \kappa(y_m^{(t)}, y_i^{(t)}) - v_0(y_i^{(t)}) \geq B_\kappa w_i^{(t)} - v_0(y_i^{(t)}), \tag{48}$$

so that

$$\begin{aligned} \forall t \in [0, \tilde{t}\,],\;\; \dot{w}_i^{(t)} &\leq -w_i^{(t)}\Big(B_\kappa w_i^{(t)} - v_0(y_i^{(t)})\Big) \\ &\leq -w_i^{(t)}\Big(B_\kappa w_i^{(t)} - B_\kappa\Big) \\ &\leq -B_\kappa w_i^{(t)}\Big(w_i^{(t)} - 1\Big), \end{aligned}$$

and, by defining $\tilde{w}_i := 1 - w_i$, we get $\dot{\tilde{w}}_i = -\dot{w}_i$, and

$$\forall t \in [0, \tilde{t}\,],\;\; -\dot{\tilde{w}}_i \leq B_\kappa \underbrace{(1 - \tilde{w}_i)}_{w_i} \underbrace{\tilde{w}_i}_{1-w_i}, \tag{49}$$

so that

$$\forall t \in [0, \tilde{t}\,],\;\; \dot{\tilde{w}}_i \geq -B_\kappa (1 - \tilde{w}_i) \tilde{w}_i. \tag{50}$$

Now, assume that there exists $t \in (0, \tilde{t}]$ such that $w_i^{(t)} \geq 1$ which is equivalent to $\tilde{w}_i^{(t)} \leq 0$. Denote $t^* = \min\{t \in [0, \tilde{t}\,] | \tilde{w}_i^{(t)} \leq 0\}$. Then, for all $t \in [0, t^*]$, we know that $1 > w_i^{(t)}$, giving

$$(B_\kappa + 1) \geq (B_\kappa + 1) w_i^{(t)}.$$

By rearranging, we get

$$\begin{aligned}\forall t \in [0, t^*],\ 1 - w_i^{(t)} &\geq B_\kappa w_i^{(t)} - B_\kappa \\ &= -B_\kappa(1 - w_i^{(t)}) \\ \tilde{w}_i^{(t)} &\geq -B_\kappa \tilde{w}_i^{(t)}\end{aligned}$$

Thus, using Grönwall's lemma, we prove that $\tilde{w}_i^{(t)} \geq \tilde{w}_i^{(0)} e^{-B_\kappa t}$ for all $t \in [0, t^*]$, and as a result $\tilde{w}_i^{(t^*)} > 0$, which is a contradiction.

Step 3: Uniform boundedness of F We recall that the kernel gradient is bounded and suppose that, for any $x \in \mathcal{X}$ and $j \in [d]$, we have $y \mapsto \partial_{y_j}\kappa(x, y)$ is bounded by B'_κ. We now show that $F : \mathbb{R}^{Md} \times [0, 1]^M \to \mathbb{R}^{M(d+1)}$ is a bounded function. Indeed, assuming that $z \in \mathbb{R}^{Md} \times [0, 1]^M$, we know that

$$\begin{aligned}|F_{y,i}(z)_j| &= \alpha \left| \sum_{m=1}^{M} w_m \nabla_{2,j}\kappa(y_m, y_i) - \partial_j v_0(y_m) \right| && (51)\\ &\leq \alpha \sum_{m=1}^{M} w_m B'_\kappa + B'_\kappa && (52)\\ &= 2\alpha M B'_\kappa. && (53)\end{aligned}$$

Similarly,

$$\begin{aligned}|F_{w,i}(z)| &= w_i \left| \sum_{m=1}^{M} w_m \kappa(y_m, y_i) - v_0(y_m) \right| && (54)\\ &\leq \sum_{m=1}^{M} B_\kappa + B_\kappa && (55)\\ &= 2MB_\kappa. && (56)\end{aligned}$$

Therefore, F is uniformly bounded on the domain $\mathbb{R}^{Md} \times [0, 1]^M$.

Step 4: Extension of the solution to $[0, +\infty)$ Suppose that we set $z_0 \in \mathbb{R}^{Md} \times (0, 1)^M$ and consider when

$$\begin{cases} \dot{z} &= F(z) \\ z(0) &= z_0 \end{cases} \tag{57}$$

admits a maximal solution on an interval $[0, T_{\max})$ for some positive $T_{\max} < +\infty$. Then, the set

$$S_{\max} := \overline{\{\hat{z}(t); t \in [0, T_{\max})\}} \tag{58}$$

is a compact subset of $\mathbb{R}^{(d+1)M}$. Indeed, it is closed by construction and we have established that the weights are bounded by unity, independent of time. Since we work on a finite domain, we can bound any point

$$|y_{ij}| = \left| \int_0^{T_{\max}} F_{y,i}(z^{(t)})_j \,\mathrm{d}t \right| \leq 2\alpha M B'_\kappa T_{\max}.$$

Then, we know that the set $S_{\max}$ is compact. Now, let $(t_n)_{n\in\mathbb{N}^*}$ be a sequence of scalars in $[0, T_{\max})$ such that $\lim_{n\to\infty} t_n = T_{\max}$. Since $S_{\max}$ is a compact set, the sequence $\hat{z}(t_n)$ has a sub-sequence $(\hat{z}(t_{n_m}))_{m\in\mathbb{N}}$ that converges to some $z_\star \in S_{\max}$ as m goes to $+\infty$. We consider the problem

$$\begin{aligned}\dot{z} &= F(z) && (59)\\ z(0) &= z_\star. && (60)\end{aligned}$$

By using Peano existence theorem again, we prove the existence of a solution defined on a neighborhood $I_\star$ of 0. Thus we can extend $\hat{z}$ to $[0, T_{\max} + \varepsilon_\star)$, where $\varepsilon_\star > 0$ is any positive scalar such that $[0, \varepsilon_\star) \subset I_\star$, giving a contradiction. Therefore $T_{\max} = +\infty$.

Step 5: Uniqueness of solution Since we have seen that F is continuous, uniformly bounded on $\mathbb{R}^{Md} \times (0,1)^M$, assumed that the initialization gives $w^{(0)} \in (0,1)^M$, and shown that $w^{(t)} \in (0,1)^M$ for any finite $t > 0$, we know that F is a Lipschitz function on its domain and across the trajectory for all t and thus, by the Picard–Lindelöf theorem, we get that the solution is unique. We first note that this theorem gives us a convergence in a small neighborhood of zero; however, we use the same argument as in step 4 by noting that the ODE is autonomous and, thus, the time-independent bound of F provides the ability to stretch the uniqueness to time infinity. We then remark that we see the boundedness of $Y^{(t)}$ is time-dependent in step 4. Nevertheless, having only a uniform bound on our *function*, i.e., F, is sufficient to employ Picard–Lindelöf.

Step 6: MMD dissipation We remark that the WFR gradient flow chooses the time derivative of the functional first variation $\mathcal{F}_\delta$ to locally minimize the Wasserstein–Fisher–Rao (or Hellinger–Kantorovich) metric Chewi et al. (2024),

$$D_{\mathrm{WFR}}(\psi, \mu; \alpha) = \alpha \int \|\nabla \psi\|^2 \, d\mu + \int \psi^2 d\mu.$$

We can prove this using typical integration-by-parts arguments similar to, e.g., Arbel et al. (2019). Assume that $\mathcal{F}_\delta$, the first variation of our loss functional, is sufficiently smooth and recall standard calculus of variations identities

$$\frac{d}{dt}\mathcal{F}[\mu_t] = \int \left(\mathcal{F}_\delta[\mu_t] \frac{d}{dt}\mu_t \right) dy \tag{61}$$

$$= \int \mathcal{F}_\delta[\mu_t](\alpha \ \mathrm{div}(\mu_t \nabla \mathcal{F}_\delta[\mu_t]) - \mu_t \mathcal{F}_\delta[\mu_t]) dy \tag{62}$$

$$= -\left(\alpha \int \|\nabla \mathcal{F}_\delta[\mu_t]\|^2 \, d\mu_t(y) + \int \mathcal{F}_\delta[\mu_t]^2 d\mu_t \right), \tag{63}$$

where the last step uses integration-by-parts. Following Arbel et al. (2019), the MMD functional $\mathcal{F}$ exhibits the regularity needed for these calculations.

D.3 Proof of Proposition 3.3

Under Assumption 2.1, for $j \in [M]$, equation (20) can be written as

$$\sum_{i=1}^{M} w_i y_i \bar{\kappa}(y_i, y_j) = \Big(\sum_{i=1}^{M} w_i \bar{\kappa}(y_i, y_j) - \int_{\mathcal{X}} \bar{\kappa}(x, y_j) \mathrm{d}\pi(x) \Big) y_j + \int_{\mathcal{X}} x \bar{\kappa}(x, y_j) \mathrm{d}\pi(x), \tag{64}$$

which can be expressed in matrix form as (21).

D.4 Proof of Theorem 3.4

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 $Y \in \mathcal{X}_{\neq}^M$ we have

$$F_M(Y) = \frac{1}{2}\Big(C_\pi - 2\langle \hat{\boldsymbol{w}}(Y), \boldsymbol{v}_0(Y) \rangle + \langle \hat{\boldsymbol{w}}(Y), \boldsymbol{K}(Y) \hat{\boldsymbol{w}}(Y) \rangle \Big), \tag{65}$$

where $\hat{\boldsymbol{w}}(Y)$ is defined by (4) respectively, and v_0 is given by (5).

Now, using (4), we prove that

$$\langle \hat{\boldsymbol{w}}(Y), \boldsymbol{v}_0(Y) \rangle = \langle \boldsymbol{v}_0(Y), \boldsymbol{K}(Y)^{-1} \boldsymbol{v}_0(Y) \rangle = \langle \hat{\boldsymbol{w}}(Y), \boldsymbol{K}(Y) \hat{\boldsymbol{w}}(Y) \rangle. \tag{66}$$

In particular, the expression of $F_M(Y)$ simplifies further to

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

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 jth 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 $\nabla_{ij}\boldsymbol{K}(Y) \in \mathbb{R}^{M\times M}$ to be the differentiation of each element of matrix $\boldsymbol{K}(Y)$ with respect to the jth coordinate of vector y_i. Similarly, for kernel mean embedding $\boldsymbol{v}_0 : \mathcal{X}^M_{\neq} \to \mathbb{R}^M$ we denote by $\nabla_{ij}\boldsymbol{v}_0$ the vector obtained by differentiating each element of $\boldsymbol{v}_0$ with respect to the jth coordinate of vector y_i. In other words, we have

$$\nabla_{ij} F_M = \frac{\partial}{\partial y_{ij}} F_M(Y), \quad \nabla_{ij}\boldsymbol{K}(Y) = \frac{\partial}{\partial y_{ij}}\boldsymbol{K}(Y), \quad \nabla_{ij}\boldsymbol{v}(Y) = \frac{\partial}{\partial y_{ij}}\boldsymbol{v}(Y) \tag{67}$$

Using (66) and matrix calculus, we have

$$2\nabla_{ij} F_M(Y) = -2\langle \hat{\boldsymbol{w}}(Y),\, \boldsymbol{K}(Y)\nabla_{ij}\hat{\boldsymbol{w}}(Y)\rangle - \langle \hat{\boldsymbol{w}}(Y),\, \nabla_{ij}\boldsymbol{K}(Y)\,\hat{\boldsymbol{w}}(Y)\rangle. \tag{68}$$

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

$$\nabla_{ij}\hat{\boldsymbol{w}}(Y) = \boldsymbol{K}(Y)^{-1}\nabla_{ij}\boldsymbol{v}_0(Y) - \boldsymbol{K}(Y)^{-1}\,\nabla_{ij}\boldsymbol{K}(Y)\,\boldsymbol{K}(Y)^{-1}\boldsymbol{v}_0(Y), \tag{69}$$

$$\boldsymbol{K}(Y)\nabla_{ij}\hat{\boldsymbol{w}}(Y) = \nabla_{ij}\boldsymbol{v}_0(Y) - \nabla_{ij}\boldsymbol{K}(Y)\,\hat{\boldsymbol{w}}(Y) \tag{70}$$

so that

$$\langle \hat{\boldsymbol{w}}(Y),\ \boldsymbol{K}(Y)\nabla_{ij}\hat{\boldsymbol{w}}(Y)\rangle = \langle \hat{\boldsymbol{w}}(Y),\ \nabla_{ij}\boldsymbol{v}_0(Y)\rangle - \langle \hat{\boldsymbol{w}}(Y),\ (\nabla_{ij}\boldsymbol{K}(Y))\hat{\boldsymbol{w}}(Y)\rangle. \tag{71}$$

Therefore, we have

$$\begin{aligned} 2\nabla_{ij} F_M(Y) &= -\langle \hat{\boldsymbol{w}}(Y),\, (\nabla_{ij}\boldsymbol{K}(Y))\,\hat{\boldsymbol{w}}(Y)\rangle + \\ &\quad -\ 2\,(\,\langle \hat{\boldsymbol{w}}(Y),\, \nabla_{ij}\boldsymbol{v}_0(Y)\rangle - \langle \hat{\boldsymbol{w}}(Y),\, (\nabla_{ij}\boldsymbol{K}(Y))\,\hat{\boldsymbol{w}}(Y)\rangle\,) \\ &= -2\langle \hat{\boldsymbol{w}}(Y),\, \nabla_{ij}\boldsymbol{v}_0(Y)\rangle + \langle \hat{\boldsymbol{w}}(Y),\, (\nabla_{ij}\boldsymbol{K}(Y))\,\hat{\boldsymbol{w}}(Y)\rangle. \end{aligned} \tag{72}$$

Now recall that

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

for all m_1, $m_2 \in [M]$. Therefore, under Assumption 2.1,

$$\nabla_{ij}\kappa(y_{m_1}, y_{m_2}) = \delta_{i,m_1}(y_{m_2 j} - y_{ij})\bar{\kappa}(y_i, y_{m_2}) + \delta_{i,m_2}(y_{m_1 j} - y_{ij})\bar{\kappa}(y_{m_1}, y_i). \tag{73}$$

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

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

where $\boldsymbol{e}_i$ is the ith elementary vector, i.e., $[\boldsymbol{e}_i]_m = \delta_{im}$, and we define

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

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

$$\begin{aligned} \langle \hat{\boldsymbol{w}}(Y),\, (\nabla_{ij}\boldsymbol{K}(Y))\,\hat{\boldsymbol{w}}(Y)\rangle &= \langle \hat{\boldsymbol{w}}(Y),\, (\boldsymbol{b}_{ij}\otimes\boldsymbol{e}_i + \boldsymbol{e}_i\otimes\boldsymbol{b}_{ij})\hat{\boldsymbol{w}}(Y)\rangle \\ &= 2\hat{w}_i(Y)\langle \hat{\boldsymbol{w}}(Y),\, \boldsymbol{b}_{ij}\rangle \\ &= 2\hat{w}_i(Y)\Big(\langle \hat{\boldsymbol{w}}(Y),\, Y_j\odot\bar{\boldsymbol{K}}_i - y_{ij}\bar{\boldsymbol{K}}_i\rangle\Big) \\ &= 2\hat{w}_i(Y)\Big(\langle \hat{\boldsymbol{w}}(Y),\, Y_j\odot\bar{\boldsymbol{K}}_i\rangle - y_{ij}\langle \hat{\boldsymbol{w}}(Y),\, \bar{\boldsymbol{K}}_i\rangle\Big) \end{aligned} \tag{74}$$

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

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

where $x_j \in \mathbb{R}$ is the jth entry of integration variable $x \in \mathbb{R}^d$, and we recall that $\bar{v}_0(y) = \int_{\mathcal{X}} \bar{\kappa}(x, y)\, \mathrm{d}\pi(x)$. We then have

$$\nabla_{ij} \boldsymbol{v}_0(Y) = \left([\bar{v}_1(y_i)]_j - y_{ij}\bar{v}_0(y_i)\right) \boldsymbol{e}_i.$$

Thus,

$$\langle \hat{\boldsymbol{w}}(Y),\, \nabla_{ij}\boldsymbol{v}_0(Y)\rangle = \hat{w}_i(Y)\left([\bar{v}_1(y_i)]_j - y_{ij}\bar{v}_0(y_i)\right). \tag{75}$$

Combining (72), (74) and (75), we obtain

$$\begin{aligned} 2\nabla_{ij}F_M(Y) &= -2\hat{w}_i(Y)\left([\bar{v}_1(y_i)]_j - y_{ij}\bar{v}_0(y_i) - \langle \hat{\boldsymbol{w}}(Y),\, Y_j \odot \bar{\boldsymbol{K}}_i\rangle + y_{ij}\langle \hat{\boldsymbol{w}}(Y),\, \bar{\boldsymbol{K}}_i\rangle\right) \\ &= -2\hat{w}_i(Y)\left([\bar{v}_1(y_i)]_j - y_{ij}\bar{v}_0(y_i) + y_{ij}\langle \hat{\boldsymbol{w}}(Y),\, \bar{\boldsymbol{K}}_i\rangle - \langle \hat{\boldsymbol{w}}(Y),\, Y_j \odot \bar{\boldsymbol{K}}_i\rangle\right) \end{aligned} \tag{76}$$

Now,

$$\begin{aligned} -y_{ij}\bar{v}_0(y_i) + y_{ij}\langle \hat{\boldsymbol{w}}(Y),\, \bar{\boldsymbol{K}}_i\rangle &= y_{ij}\left(\sum_{m=1}^{M} \hat{w}_m(Y)\bar{\kappa}(y_i, y_m) - \int_{\mathcal{X}} \bar{\kappa}(x, y_i)\,\mathrm{d}\pi(x)\right) \\ &= (\hat{\boldsymbol{v}}_1(Y))_{i,j} - [\bar{v}_1(y_i)]_j \end{aligned} \tag{77}$$

matching the definition of $\hat{\boldsymbol{v}}_1$ in Equation (22). Using the symmetry of $\bar{\kappa}$.

$$\begin{aligned} \langle \hat{\boldsymbol{w}}(Y),\, Y_j \odot \bar{\boldsymbol{K}}_i\rangle &= \sum_{m=1}^{M} \hat{w}_m(Y) y_{mj}\bar{\kappa}(y_m, y_i) \\ &= \sum_{m=1}^{M} \bar{\kappa}(y_i, y_m)\hat{w}_m(Y) y_{mj} \\ &= \langle \boldsymbol{e}_i,\, \bar{\boldsymbol{K}}(Y)\boldsymbol{W}(Y)Y_j\rangle \\ &= \left[\bar{\boldsymbol{K}}(Y)\boldsymbol{W}(Y)Y\right]_{i,j} \end{aligned} \tag{78}$$

where $\boldsymbol{W}(Y) \in \mathbb{R}^{M\times M}$ satisfies $[\boldsymbol{W}(Y)]_{i,m} = \hat{w}_i(Y)\delta_{i,m}$. Combining (77), (78), and the fact that $[\bar{v}_1(y_i)]_j$ is the (i,j)-th entry of the matrix $\bar{\boldsymbol{v}}_1(Y)$, we conclude that

$$\left([\bar{v}_1(y_i)]_j - y_{ij}\bar{v}_0(y_i) + y_{ij}\langle \hat{\boldsymbol{w}}(Y),\, \bar{\boldsymbol{K}}_i\rangle - \langle \hat{\boldsymbol{w}}(Y),\, Y_j \odot \bar{\boldsymbol{K}}_i\rangle\right)$$

is the (i,j)-entry of the matrix $\hat{\boldsymbol{v}}_1(Y) - \bar{\boldsymbol{K}}(Y)\boldsymbol{W}(Y)Y$. Thus, (76) is the (i,j)-entry of the matrix

$$2\boldsymbol{W}(Y)\left(\bar{\boldsymbol{K}}(Y)\boldsymbol{W}(Y)Y - \hat{\boldsymbol{v}}_1(Y)\right). \tag{79}$$

In other words,

$$\nabla F_M(Y) = \boldsymbol{W}(Y)\left(\bar{\boldsymbol{K}}(Y)\boldsymbol{W}(Y)Y - \hat{\boldsymbol{v}}_1(Y)\right). \tag{80}$$

When the matrices $\boldsymbol{W}(Y)$ and $\bar{\boldsymbol{K}}(Y)$ are nonsingular, we have

$$\nabla F_M(Y) = \boldsymbol{W}(Y)\bar{\boldsymbol{K}}(Y)\boldsymbol{W}(Y)\left(Y - \boldsymbol{W}(Y)^{-1}\bar{\boldsymbol{K}}(Y)^{-1}\hat{\boldsymbol{v}}_1(Y)\right). \tag{81}$$

Finally, observe that the function $\boldsymbol{\Psi}_{\mathrm{MSIP}}$ defined by (25) satisfies

$$\boldsymbol{\Psi}_{\mathrm{MSIP}}(Y) = \boldsymbol{W}(Y)^{-1}\bar{\boldsymbol{K}}(Y)^{-1}\hat{\boldsymbol{v}}_1(Y). \tag{82}$$

D.5 Proof of Proposition 3.5

Let $\mathbb{R}^d_+ = \{x \in \mathbb{R}^d \,|\, x_j > 0\, \forall j \in [M]\}$.

Lemma D.1. *The set $A = \{Y \in \mathbb{R}^{M\times d} \,|\, \det \boldsymbol{K}(Y)| > 0\}$ is open and nonempty for the squared exponential kernel given any finite bandwidth $s > 0$.*

Proof. Since $Y \mapsto \boldsymbol{K}(Y)$ and the determinant are both continuous functions, their composition $Y \mapsto \det \boldsymbol{K}(Y)$ is continuous as well. Thus, A is open as the pre-image of an open set under a continuous function. Now set $r := s \log M$ and choose Y such that $\|y_i - y_j\| > r$ for $i \neq j$ (e.g., $y_j = j(1+r)\boldsymbol{e}_1$ for elementary vector $\boldsymbol{e}_1$). We know then that, for $i \neq j$,

$$\kappa(y_i, y_j) = \exp(-\|y_i - y_j\|^2/s^2) < \exp(-r^2/s^2) \leq \frac{1}{M}.$$

Moreover, we have $\kappa(y_i, y_i) = 1$ for $i \in [M]$. Then, we use Gershgorin's circle theorem by noting the radius of every Gershgorin circle is bounded by $(M-1)/M < 1$. More precisely, this proves that the absolute value of any eigenvalue of $\boldsymbol{K}(Y)$ is bounded from below by $1/M$, and thus ensuring $\det \boldsymbol{K}(Y) > 0$. Therefore, we know that $Y \in A$ and thus $A \neq \emptyset$. ☐

Proof of Proposition 3.5. We recall that κ is the squared-exponential kernel and proceed with a simple topological argument. For any $x \in \mathbb{R}^d_+$, we define $r_x := \min_j x_j$ to describe a ball $B_{r_x} \subset \mathbb{R}^d_+$ centered on x, so we prove that $\mathbb{R}^d_+$ is an open set. Further, we use A as given in Lemma D.1 to define the open set of configurations Y with invertible kernel matrices. Finally, we know that $\boldsymbol{v}_0 \in \mathcal{H}$ and, thus, it must be continuous.

Now, recall that $\hat{\boldsymbol{w}} : A \to \mathbb{R}^M$ is defined as $\hat{\boldsymbol{w}}(Y) = \boldsymbol{K}(Y)^{-1}\boldsymbol{v}_0$, and observe that $\hat{\boldsymbol{w}}$ is continuous. Indeed, $Y \mapsto \boldsymbol{K}(Y)^{-1}$ is continuous by the continuity of $Y \mapsto \boldsymbol{K}(Y)$ and the continuity of the inverse; and $Y \mapsto \boldsymbol{v}_0(Y)$ is continuous on A, since $\boldsymbol{v}_0 \in \mathcal{H}$ and thus continuous. In the following, define $B := \{Y \in A \,|\, \hat{\boldsymbol{w}}(Y) \in \mathbb{R}^d_+\}$. Since $\mathbb{R}^d_+$ is open and $\hat{\boldsymbol{w}}$ is continuous, B is open as pre-image of an open set by a continuous function.

Consider the MSIP setting with some $Y^{(t)} \in A$ such that $\boldsymbol{w}(Y^{(t)}) \in \mathbb{R}^d_+$ and let $U^{(t)} = \boldsymbol{P}(Y^{(t)})^{-1}\nabla F_M(Y^{(t)}) = \boldsymbol{\Psi}_{\text{MSIP}}(Y^{(t)})$. We know that $\boldsymbol{P}(Y^{(t)}) = \boldsymbol{W}(Y)\boldsymbol{K}(Y)\boldsymbol{W}(Y)$ is positive definite, as $\boldsymbol{x}^\top \boldsymbol{P}(Y^{(t)})\boldsymbol{x} = (\boldsymbol{W}(Y)\boldsymbol{x})^\top \boldsymbol{K}(Y)(\boldsymbol{W}(Y)\boldsymbol{x}) > 0$ for every vector $\boldsymbol{x} \neq 0$. This comes from the fact that $\boldsymbol{W}(Y)$ is invertible and $Y \in A$, so $\boldsymbol{K}(Y)$ is positive definite, giving that both $\boldsymbol{P}(Y^{(t)})$ and $\boldsymbol{P}(Y^{(t)})^{-1}$ are positive definite. Moreover, $-\boldsymbol{P}(Y^{(t)})^{-1}\nabla F_M(Y^{(t)})$ is a descent direction of the function F_M since

$$\begin{aligned}\langle \nabla F_M(Y^{(t)}), -\boldsymbol{P}(Y^{(t)})^{-1}\nabla F_M(Y^{(t)})\rangle_F &= \\ &\sum_{i=1}^{M} \langle (\nabla F_M(Y^{(t)}))_{i,:}, -\boldsymbol{P}(Y^{(t)})^{-1}(\nabla F_M(Y^{(t)}))_{i,:}\rangle < 0\end{aligned}$$

for the Frobenius inner product $\langle \cdot, \cdot \rangle_F$. Thus, there exists some η_1 such that $F_M(Y^{(t)} - \eta_1 U^{(t)}) < F_M(Y^{(t)})$. As B is an open set, there exists $\eta_2 > 0$ such that $Y^{(t)} - \eta_2 U^{(t)} \in A$ and nonzero $\eta_3 < \eta_2$ such that $\boldsymbol{w}(Y^{(t)} - \eta_3 U^{(t)}) \in \mathbb{R}^d_+$. Step sizes η_2 and η_3 come from the openness of A and B, respectively. Thus, picking $\eta_t = \min(\eta_1, \eta_2, \eta_3)$, we get that $Y^{(t+1)} = Y^{(t)} - \eta_t U^{(t)}$ satisfies the proposition. ☐

D.6 Proof of Proposition 3.6

We first prove that there exists some bandwidth s such that any configuration of unique points drawn from empirical π satisfies $\hat{\boldsymbol{w}}(Y) \in \mathbb{R}^d_+$. First, we admit the spectral decompostion of $\boldsymbol{K}(Y)$ as $U\Sigma U^\top$ for diagonal Σ and orthogonal U. We know that

$$\boldsymbol{K}(Y)^{-1} = I - (I - \boldsymbol{K}(Y)^{-1}) = I - U(I - \Sigma^{-1})U^\top.$$

Therefore, we can see that

$$\boldsymbol{K}(Y)^{-1}\boldsymbol{v}_0(Y) = \boldsymbol{v}_0(Y) - U(I - \Sigma^{-1})U^\top \boldsymbol{v}_0(Y).$$

What remains to be shown is that we can pick some bandwidth s such that

$$\|U(I - \Sigma^{-1})U^\top \boldsymbol{v}_0(Y)\|_\infty < \|\boldsymbol{v}_0(Y)\|_\infty.$$

Using the topological equivalence bound $\|\cdot\|_\infty \leq \|\cdot\|_2$, we get that

$$\begin{aligned}\|U(I - \Sigma)U^\top \boldsymbol{v}_0(Y)\|_\infty &\leq \|U(I - \Sigma^{-1})U^\top \boldsymbol{v}_0(Y)\|_2 &(83)\\ &\leq \|(I - \Sigma^{-1})\|_2 \|\boldsymbol{v}_0(Y)\|_2 &(84)\\ &= \sqrt{M}(1 - \sigma_{\max}^{-1}), &(85)\end{aligned}$$

where $\sigma_{\max}$ is the maximum of the spectrum of $\boldsymbol{K}(Y)$, and the bound on $\|\boldsymbol{v}_0(Y)\|_2$ comes from the fact that $v_0(y) \leq 1$ (cf. (40)). Now, set $r := \min_{i \neq j} \|y_i - y_j\|_2^2$ and pick bandwidth

$$s = \frac{r}{\log(NM\sqrt{M})}.$$

Using Gershgorin's circle theorem, we know that, for any fixed $i \in [M]$,

$$\sigma_{\max} \leq \sum_{m=1}^{M} \kappa(y_i, y_m) \leq 1 + (M-1)\exp(-s^{-1}r) = 1 + \frac{M-1}{NM\sqrt{M}} < \frac{N\sqrt{M}+1}{N\sqrt{M}},$$

where we remark that this last inequality is *strict.* Then,

$$\sqrt{M}(1 - \sigma_{\max}^{-1}) < \sqrt{M}\left(1 - \frac{N\sqrt{M}}{N\sqrt{M}+1}\right) = \frac{1}{N + M^{-1/2}} < \frac{1}{N} \leq \|\boldsymbol{v}_0(Y)\|_\infty.$$

This last inequality comes from the fact that, for any x_j drawn from the support of an empirical π, we have

$$v_0(x_j) = \frac{1}{N}\sum_{i=1}^{N} \kappa(x_i, x_j) = \frac{1}{N} + \frac{1}{N}\sum_{i \neq j} \kappa(x_i, x_j) \geq \frac{1}{N}.$$

Since Y were chosen from the support of π, we then know that $N^{-1} \leq \|\boldsymbol{v}_0(Y)\|_\infty$.

D.7 Proof of Corollary 3.7

We again define $C_\pi = \int_{\mathcal{X}}\int_{\mathcal{X}} \kappa(x, x')\, \mathrm{d}\pi(x)\mathrm{d}\pi(x')$. First, for $w \in \mathbb{R}$ and $y \in \mathcal{X}$, we have

$$\mathcal{F}(w\delta_y) = \frac{1}{2}\left(C_\pi - 2wv_0(y) + w^2\kappa(y, y)\right). \tag{86}$$

Since this is a quadratic form on w, we get

$$\hat{w}(y) = \operatorname*{arg\,min}_{w \in \mathbb{R}} \mathcal{F}(w\delta_y) = \frac{v_0(y)}{\kappa(y, y)}. \tag{87}$$

Thus, by (27) in Theorem 3.4, we get

$$\begin{aligned} \nabla F_1(y) &= \frac{v_0(y)}{\kappa(y,y)}\bar{\kappa}(y,y)\frac{v_0(y)}{\kappa(y,y)}(y - \Psi_{\mathrm{MSIP}}(y)) && (88)\\ &= \frac{v_0(y)}{\kappa(y,y)}\left(\bar{\kappa}(y,y)\frac{v_0(y)}{\kappa(y,y)}y - \hat{v}_1(y)\right). && (89)\end{aligned}$$

By definition (22), we have

$$\hat{v}_1(y) = \left(\frac{v_0(y)}{\kappa(y,y)}\bar{\kappa}(y,y) - \bar{v}_0(y)\right)y + \bar{v}_1(y). \tag{90}$$

By rearranging (90), we see

$$\frac{v_0(y)}{\kappa(y,y)}\bar{\kappa}(y,y)y - \hat{v}_1(y) = \bar{v}_0(y)y - \bar{v}_1(y). \tag{91}$$

Combining (89) and (91)

$$\begin{aligned} \nabla F_1(y) &= \frac{v_0(y)}{\kappa(y,y)}\Big(\bar{v}_0(y)y - \bar{v}_1(y)\Big). && (92)\\ &= \frac{v_0(y)\bar{v}_0(y)}{\kappa(y,y)}(y - \Psi_{\mathrm{MS}}(y)) && (93)\end{aligned}$$

Now we prove that Ψ_{MS} and Ψ_{MSIP} are identical in the case that $\bar{\kappa} = \lambda\kappa$ for $\lambda \neq 0$. We first observe that

$$\bar{v}_0(y) = \mathbb{E}_\pi[\bar{\kappa}(X, y)] = \mathbb{E}_\pi[\lambda\kappa(X, y)] = \lambda v_0(y). \tag{94}$$

Then, we can see that

$$\frac{v_0(y)\bar{v}_0(y)}{\kappa(y,y)} = \frac{v_0^2(y)}{\kappa^2(y,y)}\lambda\kappa(y,y) = \frac{v_0^2(y)}{\kappa^2(y,y)}\bar{\kappa}(y,y). \tag{95}$$

Therefore, we can combine Equations (88) and (93) to arrive at the conclusion

$$\Psi_{\mathrm{MS}}(y) = \Psi_{\mathrm{MSIP}}(y) \tag{96}$$