Sampling in Unit Time with Kernel Fisher–Rao Flow

Aimee Maurais [1] **Youssef Marzouk** [1]

Abstract

We introduce a new mean-field ODE and corresponding interacting particle systems (IPS) for sampling from an unnormalized target density. The IPS are gradient-free, available in closed form, and only require the ability to sample from a reference density and compute the (unnormalized) target-to-reference density ratio. The mean-field ODE is obtained by solving a Poisson equation for a velocity field that transports samples along the geometric mixture of the two densities, $\pi_0^{1-t}\pi_1^t$, which is the path of a particular Fisher–Rao gradient flow. We employ a RKHS ansatz for the velocity field, which makes the Poisson equation tractable and enables discretization of the resulting mean-field ODE over finite samples. The mean-field ODE can be additionally be derived from a discrete-time perspective as the limit of successive linearizations of the Monge–Ampère equations within a framework known as sample-driven optimal transport. We introduce a stochastic variant of our approach and demonstrate empirically that our IPS can produce high-quality samples from varied target distributions, outperforming comparable gradient-free particle systems and competitive with gradient-based alternatives.

1. Introduction

In this work we consider the problem of *sampling via transport*: given a target distribution π_1 on $\mathbb{R}^d$ and a reference distribution π_0 on $\mathbb{R}^d$ from which we can sample, our goal is to find $T : \mathbb{R}^d \to \mathbb{R}^d$ such that $T_\sharp \pi_0 = \pi_1$, i.e., $X_0 \sim \pi_0 \Rightarrow T(X_0) \sim \pi_1$. We assume that π_0 and π_1 both admit densities and that we can evaluate the (unnormalized) *density ratio*[*] $\frac{\pi_1}{\pi_0}$ but that we do not have samples of π_1 with which to train the map, or access to gradients (including scores) of π_1 or π_0. The density ratio is available when the (unnormalized) density of π_1 is known and π_0 is chosen to be some "standard" reference (e.g., Gaussian). It is also accessible in the Bayesian setting so long as the likelihood function ℓ is known: therein π_0 is the prior distribution of some $\mathbb{R}^d$-valued parameter X and $\pi_1 \propto \ell\,\pi_0$ is the posterior distribution of X given $Y = y^*$, with $\ell(\cdot) = \pi_{Y|X}(y^*|\cdot)$. The ratio in the Bayesian setting is $\frac{\pi_1}{\pi_0} \propto \ell$, but we will use the term "likelihood" to refer to the ratio $\frac{\pi_1}{\pi_0}$ outside of the Bayesian setting as well. In applications of Bayesian inference such as data assimilation (Reich & Cotter, 2015) it is frequently the case that π_0 is only known through samples; hence, while the target density $\pi_1 \propto \pi_0\ell$ cannot be evaluated, the likelihood ℓ often can. Furthermore, in many other scientific applications, π_1 or ℓ may contain a complicated physical model whose gradients are inaccessible; hence gradient-free sampling is a necessity.

The canonical sampling approach employing the likelihood is importance sampling (Owen, 2013), which transforms an unweighted ensemble of samples of π_0 into a *weighted* ensemble, enabling the estimation of expectations under π_1. Importance sampling is the foundation for sequential Monte Carlo (SMC) methods (Del Moral et al., 2006), but is frequently plagued by issues of weight degeneracy and ensemble collapse, necessitating large ensemble sizes (Snyder et al., 2008) or interventions such as resampling (Künsch, 2005) and MCMC rejuvenation.

Alternatively, many sampling approaches use *dynamics* to define a transport incrementally, e.g., via the flow map induced by trajectories of an ODE or the stochastic mapping induced by sample paths of an SDE. In either case, the idea is to apply dynamics which will transform some initial state $X_0 \sim \pi_0$ to a state $X_S \sim \pi_{X_S} \approx \pi_1$ for some time $S > 0$. This approach underlies flow, diffusion, and bridge techniques for generative modeling, e.g., Kuang & Tabak (2019); Song et al. (2021); De Bortoli et al. (2021); Liu et al. (2022); Lipman et al. (2023); Xu et al. (2023a); Albergo et al. (2023), wherein samples from both π_0 and π_1 are almost always required for training (with Vargas et al. (2023a); Heng et al. (2024) being recent exceptions). In the setting where π_1 is known only through its unnormalized density, there are a number of dynamic sampling algorithms which have their grounding as *gradient flows* of functionals

[1]Center for Computational Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA. Correspondence to: Aimee Maurais <maurais@mit.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[*]For the remainder of this paper, the terms "density" and "density ratio" refer to unnormalized quantities unless otherwise stated.

on spaces of probability measures. There are several geometries in which one may define gradient flows on probability measures (see Chen et al. (2023) for a helpful review), but most well-known algorithms in this vein (e.g., Liu & Wang (2016); Garbuno-Inigo et al. (2020a;b); Reich & Weissmann (2021)) use some form of the Wasserstein geometry to define dynamics which must, in principle, be run for *infinite time* in order to ensure correct sampling from π_1.

In this work we develop a dynamic sampling approach based on an ODE which transports samples from π_0 to π_1 in *unit time* such that the time-dependent distribution of the samples is the geometric mixture $\pi_t \propto \pi_0^{1-t}\pi_1^t = \pi_0(\frac{\pi_1}{\pi_0})^t$, $t \in [0,1]$. Although our algorithms are gradient-free and only require the likelihood $\frac{\pi_1}{\pi_0}$, the path of distributions π_t corresponds to the Fisher–Rao gradient flow of the expected negative log likelihood. The underlying dynamics are described by a **mean-field ODE model**, which we show is the limit of two different **interacting particle systems**. These interacting particle systems, which we generally refer to as *Kernel Fisher–Rao Flow*, are obtained in two distinct but related ways. On one hand, in continuous time, the mean-field ODE can be obtained from the weak formulation of a Poisson equation for a *velocity field* defined in a reproducing kernel Hilbert space (RKHS), from which we obtain a finite-particle ODE system by approximating expectations via Monte Carlo (Section 3). On the other, in discrete time, one can approximate the optimal transport *map* which pushes π_t to $\pi_{t+\Delta t}$ via linearization of the Monge–Ampère equations discretized over finitely many kernel basis functions and finitely many samples (Section 4). This linearization yields an interacting particle system which in discrete time is distinct from, but in continuous time identical to, that obtained via the RKHS approach to Poisson's equation.

2. Background

Sampling via measure transport is an active area of research, with many computational approaches (Marzouk et al., 2016; Kobyzev et al., 2020; Papamakarios et al., 2021; Trillos et al., 2023) appearing in recent years. Most practical transport maps are parameterized, and thus a crucial part of realizing them is selecting an appropriately rich function class within which to search for the map. Common map approximation classes include polynomials (Jaini et al., 2019), radial basis functions (Spantini et al., 2022), composed simple transformations (Rezende & Mohamed, 2015; Kobyzev et al., 2020; Papamakarios et al., 2021), neural networks (Bunne et al., 2022; Taghvaei & Hosseini, 2022; Baptista et al., 2023a), and reproducing kernel Hilbert spaces (Liu & Wang, 2016; Kuang & Tabak, 2019; Katzfuss & Schäfer, 2023). Determining an appropriate basis to represent a transport map can be challenging, especially when the target and reference distributions are high-dimensional or differ from each other considerably. For this reason it may be necessary to employ, e.g., adaptive feature selection algorithms (Baptista et al., 2023b) or dimension reduction techniques (Spantini et al., 2018; Chen et al., 2019; Brennan et al., 2020; Dai & Seljak, 2021).

As an alternative to searching for a single, potentially highly complex transport map which pushes the reference π_0 directly to the target π_1, one can instead prescribe a *path* of distributions $(\pi_t)_{t\in[0,1]}$ having the target and reference as endpoints and seek a sequence of maps $T_1, \dots, T_N$ which push samples along a discretization of the path, as depicted in Figure 1. The composed map $T = T_N \circ T_{N-1} \circ \cdots \circ T_1$ pushes forward π_0 to π_1. In continuous time this approach becomes one of finding a *velocity field* $v_t : \mathbb{R}^d \to \mathbb{R}^d$ such that the solution to the initial value problem

$$\dot{X}_t = v_t(X_t), \quad X_0 \sim \pi_0$$

has distribution π_t. Samplers of unnormalized densities which employ this homotopy approach frequently take π_t to be the geometric mixture

$$\pi_t \propto \pi_0^{1-t}\pi_1^t = \pi_0 \left(\frac{\pi_1}{\pi_0}\right)^t, \quad t \in [0,1], \tag{1}$$

which interpolates between π_0 and π_1 in unit time. This mixture may be referred to as the "power posterior" path and appears, for example, in annealed importance sampling (Neal, 2001; Brekelmans et al., 2020; Korba & Portier, 2022; Goshtasbpour et al., 2023) and parallel tempering (Geyer, 1991; Earl & Deem, 2005; Syed et al., 2021). In Bayesian computation, this path is sometimes referred to as "tempered likelihood" and has been used as the basis for algorithms which generate (approximate) posterior samples (Reich, 2011; Daum & Huang, 2013; Iglesias et al., 2013; Ding & Li, 2021) or posterior densities (Dia, 2023).

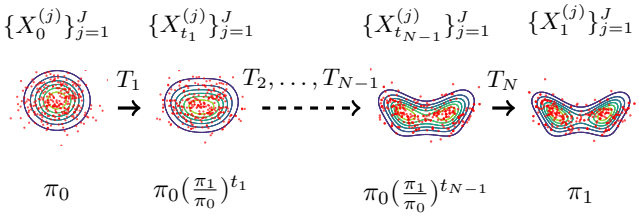

Figure 1. We employ a homotopy-based sampling scheme in this work, deriving a mean-field ODE which approximately transports a reference π_0 to a target π_1 in unit time. In discrete time this approach amounts to obtaining incremental maps $T_1, \dots, T_N$.

2.1. Related Work

The idea of using dynamics with the tempered likelihood to sample the posterior distribution in Bayesian inference was, to our knowledge, first developed by Daum & Huang

(2011) and Reich (2011) in the context of filtering. In Daum & Huang (2011; 2013) and related works by the same authors, the tempered likelihood is used to derive ODEs and SDEs for nonlinear filtering in full generality, and as such these systems require gradients and even Hessians of the likelihood and prior. The algorithms for propagating samples from prior to posterior along the tempered likelihood in Reich (2011) are of ensemble-Kalman type and employ a Gaussian approximation for each π_t, meaning that their expressivity is limited. A similar methodology to Reich (2011), now known as Ensemble Kalman Inversion (EKI), was proposed for computing point estimates in Bayesian inverse problems in Iglesias et al. (2013). The iteration underlying EKI is run for infinite time, but, as noted by, e.g., Ding & Li (2021), when the iteration is stopped at $t = 1$ samples from an approximation to the posterior are obtained. Even in the limit of infinite particles and continuous time, however, ensemble Kalman methods are not consistent samplers for general (i.e., non-Gaussian) posteriors.

The tempered likelihood path is frequently employed in sequential Monte Carlo (SMC) methods, which rely on a series of interwoven importance re-sampling and mutation steps to gradually transform samples from π_0 into approximate samples from π_1. Several recent works have sought to use transport within SMC to either replace or reduce the frequency of multinomial resampling. The basic idea, which was introduced in Reich (2013) and motivates our development in Section 4, is to use importance weights to obtain a transport between π_t and $\pi_{t+\Delta t}$. In Ruchi et al. (2019; 2021); Myers et al. (2021), discrete optimal transport couplings are used to define linear transformations which replace the importance resampling steps typically employed in SMC. In a similar vein, the annealed flow transport Monte Carlo of Arbel et al. (2021) uses transport as a preconditioner for SMC; on each SMC iteration a parametric transport map between π_t and $\pi_{t+\Delta t}$ is learned and applied to samples from π_t before the standard resampling and mutation steps are performed. In each of these works the transport steps remain embedded within SMC schemes, while our algorithms are based on dynamics and do not require resampling or application of mutation kernels.

Concurrently with our work, Wang & Nüsken (2024) developed similar mean-field ODE systems and algorithms for transporting samples along the tempered likelihood. Their algorithms are motivated from a kernel mean embedding perspective, rather than one of optimal transport and Fisher–Rao gradient flows, and can be made to match ours by specifying particular choices of the parameters v_t^0 and C_t in their setup. In our work we introduce both deterministic and stochastic interacting particle systems for sampling the tempered likelihood, but the algorithms considered in Wang & Nüsken (2024) are purely deterministic.

Notation We use $K(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ to represent a symmetric positive definite kernel on $\mathbb{R}^d$ and denote by $(\mathcal{H}_K, \langle\cdot,\cdot\rangle_{\mathcal{H}_K})$ the reproducing kernel Hilbert space (Steinwart & Christmann, 2008) associated with K. We assume that $K(\cdot, x)$ is C^2 and use $\nabla_1 K(\cdot,\cdot)$ to refer to the gradient of K with respect to the first argument. $\mathcal{P}_{\mathrm{ac}}(\mathbb{R}^d)$ denotes the space of probability measures on $\mathbb{R}^d$ which admit densities.

3. Methodology: Poisson Equation in Reproducing Kernel Hilbert Space

Our goal is to find a time-varying velocity field $v_t : \mathbb{R}^d \to \mathbb{R}^d$ such that the distribution of X_t evolving according to

$$\dot{X}_t = v_t(X_t), \quad X_0 \sim \pi_0 \tag{2}$$

is the geometric mixture (1). Had we access to such a velocity field, we could obtain samples from π_1 by sampling π_0 and simulating the dynamics (2) for unit time. It can be shown that π_t in (1) satisfies

$$\partial_t \pi_t = \pi_t \left(\log \tfrac{\pi_1}{\pi_0} - \mathbb{E}_{\pi_t}\left[\log \tfrac{\pi_1}{\pi_0}\right] \right),$$

which is the Fisher–Rao gradient flow of the functional $\mathcal{F} : \mathcal{P}_{\mathrm{ac}}(\mathbb{R}^d) \to \mathbb{R}$ defined as

$$\mathcal{F}(\mu) = -\mathbb{E}_{\mu}\left[\log \tfrac{\pi_1}{\pi_0}\right].$$

$\mathcal{F}(\mu)$ is the expected negative log likelihood under μ. By the continuity equation, a velocity field in (2) yielding $X_t \sim \pi_t$ must then satisfy

$$-\nabla \cdot (\pi_t v_t) = \pi_t \left(\log \tfrac{\pi_1}{\pi_0} - \mathbb{E}_{\pi_t}\left[\log \tfrac{\pi_1}{\pi_0}\right] \right). \tag{3}$$

There are many possible solutions to the PDE (3), but if, as in Taghvaei & Mehta (2023); Reich (2011), we insist that in the limit $\Delta t \to 0$ the expected transportation cost $\frac{1}{\Delta t^2}\mathbb{E}_{\pi_t}[\|X_t - X_{t+\Delta t}\|^2]$ is minimized for each t, we obtain a constrained optimization problem for each v_t with a unique solution,

$$\min_{v_t : \mathbb{R}^d \to \mathbb{R}^d} \int_{\mathbb{R}^d} \|v_t\|^2 \, \mathrm{d}\pi_t$$
$$\text{s.t. } -\nabla \cdot (\pi_t v_t) = \pi_t \left(\log \tfrac{\pi_1}{\pi_0} - \mathbb{E}_{\pi_t}\left[\log \tfrac{\pi_1}{\pi_0}\right] \right).$$

It can be shown using the geometry of optimal transport (e.g., Chewi (2023, Theorem 1.3.19)) or calculus of variations (Reich, 2011) that the solution to this problem is $v_t = \nabla u_t$, where u_t satisfies the Poisson equation

$$-\nabla \cdot (\pi_t \nabla u_t) = \pi_t \left(\log \tfrac{\pi_1}{\pi_0} - \mathbb{E}_{\pi_t}\left[\log \tfrac{\pi_1}{\pi_0}\right] \right). \tag{4}$$

We make (4) tractable by searching for u_t in the RKHS $\mathcal{H}_K$, i.e., taking $u_t(\cdot) = \int_{\mathbb{R}^d} K(\cdot, x) f_t(x) \, \mathrm{d}\pi_t(x)$ for some

$f_t : \mathbb{R}^d \to \mathbb{R}$, and enforcing the weak form of (4), as in Laugesen et al. (2015), for kernel test functions $K(\cdot, x)$,

$$\int_{\mathbb{R}^d} \langle \nabla_1 K(y,x),\ \nabla u_t(y) \rangle \,\mathrm{d}\pi_t(y) = \int_{\mathbb{R}^d} K(y,x) \left(\log \tfrac{\pi_1}{\pi_0}(y) - \mathbb{E}_{\pi_t}\left[\log \tfrac{\pi_1}{\pi_0}\right] \right) \mathrm{d}\pi_t(y). \quad (5)$$

We require (5) to hold for all $x \in \mathbb{R}^d$. Substituting the form of u_t into (5), we have

$$\iint_{\mathbb{R}^d \times \mathbb{R}^d} f_t(z) \,\langle \nabla_1 K(y,\cdot),\ \nabla_1 K(y,z) \rangle \,\mathrm{d}\pi_t(y)\,\mathrm{d}\pi_t(z) = \int_{\mathbb{R}^d} K(\cdot, y) \left(\log \tfrac{\pi_1}{\pi_0}(y) - \mathbb{E}_{\pi_t}\left[\log \tfrac{\pi_1}{\pi_0}\right] \right) \mathrm{d}\pi_t(y). \quad (6)$$

We write the relationship (6) succinctly as $M_{\pi_t} f_t(x) = K_{\pi_t}(\log \frac{\pi_1}{\pi_0} - \mathbb{E}_{\pi_t}[\log \frac{\pi_1}{\pi_0}])(x)$, where the integral operator M_{π_t} maps functions $g : \mathbb{R}^d \to \mathbb{R}$ to $M_{\pi_t} g(\cdot) =$

$$\int_{\mathbb{R}^d} g(z) \mathbb{E}_{\pi_t}[\langle \nabla_1 K(X_t,\cdot),\ \nabla_1 K(X_t,z) \rangle] \,\mathrm{d}\pi_t(z) = \iint_{\mathbb{R}^d \times \mathbb{R}^d} g(z) \langle \nabla_1 K(y,\cdot),\ \nabla_1 K(y,z) \rangle \,\mathrm{d}\pi_t(y)\mathrm{d}\pi_t(z)$$

and the kernel integral operator K_{π_t} maps g to $K_{\pi_t} g(\cdot) = \int_{\mathbb{R}^d} g(z) K(\cdot, z) \,\mathrm{d}\pi_t(z)$. Under the condition that M_{π_t} is invertible, f_t is given by

$$f_t = M_{\pi_t}^{-1} K_{\pi_t} \left(\log \tfrac{\pi_1}{\pi_0} - \mathbb{E}_{\pi_t}\left[\log \tfrac{\pi_1}{\pi_0}\right] \right)$$

and we have $v_t(\cdot) = \nabla u_t(\cdot) =$

$$\int_{\mathbb{R}^d} \nabla_1 K(\cdot, x) M_{\pi_t}^{-1} K_{\pi_t} \left(\log \tfrac{\pi_1}{\pi_0} - \mathbb{E}_{\pi_t}\left[\log \tfrac{\pi_1}{\pi_0}\right] \right)(x) \,\mathrm{d}\pi_t(x). \quad (7)$$

Therefore, starting from $X_0 \sim \pi_0$, the **mean-field ODE**

$$\dot{X}_t = v_t(X_t) = \mathbb{E}_{\rho_t}\left[\nabla_1 K(X_t, X') M_{\rho_t}^{-1} K_{\rho_t} \left(\log \tfrac{\pi_1}{\pi_0} - \mathbb{E}_{\rho_t}\left[\log \tfrac{\pi_1}{\pi_0}\right] \right)(X') \right], \quad (8)$$

can be used to evolve samples from π_0 such that $\rho_t = \mathrm{Law}(X_t)$ is approximately $\pi_t \propto \pi_0^{1-t}\pi_1^t$, and hence at $t = 1$ they are approximately distributed as π_1.

We note that the potential u_t obtained by solving a weak-form Poisson equation over the RKHS,

$$u_t = \int_{\mathbb{R}^d} K(\cdot, x) M_{\pi_t}^{-1} K_{\pi_t} \left(\log \tfrac{\pi_1}{\pi_0} - \mathbb{E}_{\pi_t}\left[\log \tfrac{\pi_1}{\pi_0}\right] \right)(x) \,\mathrm{d}\pi_t(x)$$

is only an approximation of the solution of (4), since we are enforcing the weak form (5) for a specific (limited) class of test functions. For this reason we distinguish between the target path of distributions $\pi_t \propto \pi_0^{1-t}\pi_1^t$ and $\rho_t = \mathrm{Law}(X_t)$. Understanding the difference between these paths and its dependence on the choice of RKHS $\mathcal{H}_K$ is an important area for future work. Empirically we find the quality of samples generated by running a discretization of (8) to be good; see Section 6.

Because the quantities appearing in v_t can be written as expectations with respect to ρ_t, the mean-field model (8) can be approximated for finite samples as an **interacting particle system** (IPS),

$$\dot{X}_t^{(j)} = \left(\nabla_1 K(X_t^{(j)}, X_t^{(1)}) \ \cdots\ \nabla_1 K(X_t^{(j)}, X_t^{(J)}) \right) M_t^{-1} \cdot \frac{1}{J}\sum_{k=1}^{J} \left(\log \tfrac{\pi_1}{\pi_0}(X_t^{(k)}) - \frac{1}{J}\sum_{i=1}^{J} \log \tfrac{\pi_1}{\pi_0}(X_t^{(i)}) \right) \begin{pmatrix} K(X_t^{(k)}, X_t^{(1)}) \\ \vdots \\ K(X_t^{(k)}, X_t^{(J)}) \end{pmatrix}, \quad (9)$$

with $j \in \{1, \ldots, J\}$, $t \in [0,1]$, $\{X_0^{(j)}\}_{j=1}^J \overset{\text{i.i.d.}}{\sim} \pi_0$, and $M_t \in \mathbb{R}^{J \times J}$ given by

$$(M_t)_{\ell,m} = \frac{1}{J}\sum_{i=1}^{J} \langle \nabla_1 K(X_t^{(i)}, X_t^{(\ell)}),\, \nabla_1 K(X_t^{(i)}, X_t^{(m)}) \rangle, \qquad \ell, m = 1, \ldots, J.$$

This IPS (9) is obtained by approximating the expectations in (8) with Monte Carlo; see Appendix A.1 for a detailed derivation. We refer to (9) as **Kernel Fisher–Rao Flow (KFRFlow)** and offer a few observations:

- We only require the ability to sample π_0 and compute the log-density ratio $\log \frac{\pi_1}{\pi_0}$ in order to simulate the ODE (9). In particular contrast to Stein variational gradient descent (SVGD), unadjusted or Metropolis-adjusted Langevin samplers (ULA and MALA), and some recent Langevin-based Bayesian inference approaches (Garbuno-Inigo et al., 2020b; Reich & Weissmann, 2021), we do not require gradients or scores of π_0 or π_1. Furthermore, quantities of the form $\log \frac{\pi_1}{\pi_0} - \mathbb{E}_\rho[\log \frac{\pi_1}{\pi_0}]$ appearing in Equations (8) and (9) are invariant under scaling of $\frac{\pi_1}{\pi_0}$ by constant factors. Thus, we do not require knowledge of the normalizing constants of π_1 or π_0 to apply KFRFlow.
- KFRFlow is a simple, closed-form ODE and does not require use of numerical optimization to estimate a score or velocity field. This feature stands in contrast to many comparable finite-time dynamic approaches for sampling from unnormalized densities, including normalizing flows (Rezende & Mohamed, 2015), diffusion or score-based models (Heng et al., 2024; Vargas et al., 2023a), and recent approaches which also employ the tempered likelihood (Vargas et al., 2023b; Tian et al., 2024).

- Similarly to SVGD (Liu & Wang, 2016), which can be viewed as following a Wasserstein gradient flow with a kernelized ODE, KFRFlow is deterministic and can be viewed as following a Fisher–Rao gradient flow with a kernelized ODE.

- M_t is the finite-particle analogue of M_{π_t}. A necessary condition for invertibility of M_t, $J \leq dJ$, is satisfied by construction in the IPS (9). This fact can be seen by noting that $M_t = \frac{1}{J}\sum_{i=1}^{J} \nabla\mathbf{K}_t(X_t^{(i)})\nabla\mathbf{K}_t(X_t^{(i)})^\top$, where $\mathbf{K}_t : \mathbb{R}^d \to \mathbb{R}^J$ is the concatenation $\mathbf{K}_t(\cdot) = (K(\cdot, X_t^{(1)}), \ldots, K(\cdot, X_t^{(J)}))^\top$ and $\nabla\mathbf{K}_t \in \mathbb{R}^{J\times d}$ is the Jacobian of $\mathbf{K}_t$.

Although it is intriguing to view the mean-field ODE model (8) as resulting from kernelization of a Fisher–Rao gradient flow, we can recover it separately as the limit of a *discrete-time* interacting particle system obtained using sample-driven optimal transport (Kuang & Tabak, 2019). We discuss this perspective in the following section.

4. Discrete-Time Interpretation: Sample-Driven Optimal Transport

In discrete time, the problem of finding a velocity field v_t such that the flow $\dot{X}_t = v_t(X_t)$ has distribution $\pi_t \propto \pi_0^{1-t}\pi_1^t$ becomes one of finding transport maps $T_1, \ldots, T_N$ which push samples from π_0 along a discretization of π_t. While we we can obtain such maps by discretizing the IPS (9), for example taking $X_{t+\Delta t} = X_t + \Delta t \cdot v_t(X_t)$, we can alternately search for the maps *directly* via a framework introduced as sample-driven optimal transport in Trigila & Tabak (2016); Kuang & Tabak (2019), modified for our setting in which target samples are unavailable.

Suppose that at time $t \in [0, 1)$ we have samples $\{X_t^{(j)}\}_{j=1}^J \sim \pi_t$ which we would like to push forward to $\pi_{t+\Delta t} \propto \pi_t(\frac{\pi_1}{\pi_0})^{\Delta t}$. Given that π_t and $\pi_{t+\Delta t}$ both admit densities, there are many maps $T : \mathbb{R}^d \to \mathbb{R}^d$ satisfying $T_\sharp\pi_t = \pi_{t+\Delta t}$. The optimal transport approach (Villani, 2021), which we will approximate, is to seek the map which minimizes expected transport cost,

$$\min_{T_\sharp\pi_t = \pi_{t+\Delta t}} \mathbb{E}_{\pi_t}[\|T(X_t) - X_t\|^2]. \tag{10}$$

Owing to the choice of quadratic cost, it can be shown that the optimal map in (10) is the unique convex gradient which pushes forward π_t to $\pi_{t+\Delta t}$ (Brenier, 1991). That is, if we find $T = \nabla\phi$ satisfying $T_\sharp\pi_t = \pi_{t+\Delta t}$ with $\phi : \mathbb{R}^d \to \mathbb{R}$ convex, we have found the optimal transport map. Thus, we can search for the optimal transport map by seeking $\phi : \mathbb{R}^d \to \mathbb{R}$ convex such that $\nabla\phi_\sharp\pi_t = \pi_{t+\Delta t}$. The push-forward condition $\nabla\phi_\sharp\pi_t = \pi_{t+\Delta t}$ can be written as a Monge–Ampère PDE (Evans, 1997)

$$\pi_{t+\Delta t}(\nabla\phi(x)) \det(H_\phi(x)) = \pi_t(x),$$

where H_ϕ is the Hessian of ϕ, and interpreted in weak form as for all $f : \mathbb{R}^d \to \mathbb{R}$ continuous

$$\int_{\mathbb{R}^d} f(\nabla\phi(x))\,\mathrm{d}\pi_t(x) = \int_{\mathbb{R}^d} f(y)\,\mathrm{d}\pi_{t+\Delta t}(y). \tag{11}$$

Given that we only have finitely many samples of π_t and access to the ratio $\frac{\pi_1}{\pi_0}$, we arguably do not have enough information to find a map $T = \nabla\phi$ which exactly satisfies $T_\sharp\pi_t = \pi_{t+\Delta t}$. Thus we apply a Galerkin approximation to (11) over a basis of kernel test functions located at each particle, $\{K(\cdot, X_t^{(j)}) : j = 1, \ldots, J\}$, seeking a map

$$\nabla\phi_{\mathbf{s}}(x) = x + \sum_{j=1}^{J} s_j \nabla_1 K(x, X_t^{(j)}), \tag{12}$$

and discretizing the weak form (11) with

$$\int_{\mathbb{R}^d} K(\nabla_{\mathbf{s}}\phi(x), X_t^{(j)})\,\mathrm{d}\pi_t(x) = \int_{\mathbb{R}^d} K(y, X_t^{(j)})\,\mathrm{d}\pi_{t+\Delta t}(y), \tag{13}$$

$j = 1, \ldots, J$. Approximating (13) via Monte Carlo, we seek map coefficients $\mathbf{s} = (s_1, \ldots, s_m)$ such that

$$\sum_{j=1}^{J} \tfrac{1}{J}\mathbf{K}_t(\nabla\phi_{\mathbf{s}}(X_t^{(j)})) = \sum_{j=1}^{J} w_t^{(j)}\mathbf{K}_t(X_t^{(j)}), \tag{14}$$

where the $w_t^{(j)}$ are self-normalized importance weights,

$$w_t^{(j)} = \frac{(\frac{\pi_1}{\pi_0}(X_t^{(j)}))^{\Delta t}}{\sum_{i=1}^{J}(\frac{\pi_1}{\pi_0}(X_t^{(i)}))^{\Delta t}}, \quad j = 1, \ldots, J$$

and $\mathbf{K}_t(\cdot) = (K(\cdot, X_t^{(1)}), \ldots, K(\cdot, X_t^{(J)}))^\top$ is as before. Kuang & Tabak (2019) refer to the relationship (14) as *sample equivalence* and denote it by $\{\nabla\phi_{\mathbf{s}}(X_t^{(j)})\}_{j=1}^J \sim \{w_t^{(j)}X_t^{(j)}\}_{j=1}^J$. Because we have discretized the Monge–Ampère equations (11) over finite samples and feature functions, a solution $\mathbf{s}$ to (14) is not guaranteed to yield a unique or optimal map. The *sample-driven* OT problem then, as formulated in Kuang & Tabak (2019), is to find a minimum cost map $\nabla\phi_{\mathbf{s}}$ which satisfies sample-equivalence,

$$\min_{\{\nabla\phi_{\mathbf{s}}(X_t^{(j)})\}_{j=1}^J \sim \{w_t^{(j)}X_t^{(j)}\}_{j=1}^J} \sum_{j=1}^{J} \left\| X_t^{(j)} - \nabla\phi_{\mathbf{s}}(X_t^{(j)}) \right\|^2. \tag{15}$$

Returning to (14), admissible choices of $\mathbf{s}$ in (15) can be identified via root-finding: denote by $\mathbf{a}$ and $\mathbf{b}$ the means of $\mathbf{K}_t$ over the unweighted and weighted reference ensembles,

$$\mathbf{a} = \frac{1}{J}\sum_{j=1}^{J} \mathbf{K}_t(X_t^{(j)}), \quad \mathbf{b} = \sum_{j=1}^{J} w_t^{(j)}\mathbf{K}_t(X_t^{(j)}) \in \mathbb{R}^J.$$

For $\mathbf{s} \in \mathbb{R}^J$, define $G : \mathbb{R}^J \to \mathbb{R}^J$ to be the sample mean of $\mathbf{K}_t$ over $\{\nabla\phi_{\mathbf{s}}(X_t^{(j)})\}_{j=1}^J$,

$$G(\mathbf{s}) = \frac{1}{J}\sum_{j=1}^{J} \mathbf{K}_t(X_t^{(j)} + \nabla\mathbf{K}_t(X_t^{(j)})^\top \mathbf{s}).$$

In order for sample-equivalence to be satisfied, we need to find $\mathbf{s}^*$ such that $G(\mathbf{s}^*) = \mathbf{b}$.

Kuang & Tabak (2019) demonstrate that if the Jacobian of G at $\mathbf{s} = 0$

$$\nabla G(\mathbf{s})|_{\mathbf{s}=0} = \frac{1}{J}\sum_{i=1}^{J} \nabla\mathbf{K}_t(X_t^{(i)})\nabla\mathbf{K}_t(X_t^{(i)})^\top \equiv M_t$$

is nonsingular (for which the necessary condition $J \leq dJ$ is automatically satisfied), G is a bijection from a neighborhood U about $\mathbf{s} = \mathbf{0}$ to a neighborhood V about $G(\mathbf{0}) = \mathbf{a}$. If $\mathbf{b} \in V$, then the potential $\phi_{\mathbf{s}}$ parameterized with $\mathbf{s}^* = G^{-1}(\mathbf{b})$ gives the *global minimum* of the sample-based OT problem (15) restricted to maps of the form (12). Furthermore, Kuang & Tabak (2019) show that if the kernels are C^2, then $\phi_{\mathbf{s}^*}$ is locally convex.

For sufficiently small Δt, the system $G(\mathbf{s}^*) = \mathbf{b}$ (14) will be close to linear. Thus, for the sake of efficiency we may approximate $\mathbf{s}^*$ with a single Newton step, setting

$$\mathbf{s}^* \approx -\left(\frac{1}{J}\sum_{i=1}^{J} \nabla\mathbf{K}_t(X_t^{(i)})\nabla\mathbf{K}_t(X_t^{(i)})^\top\right)^{-1} \cdot \sum_{k=1}^{J}(\tfrac{1}{J} - w_t^{(k)})\mathbf{K}_t(X_t^{(k)}), \quad (16)$$

to arrive at the update

$$X_{t+\Delta t}^{(j)} = X_t^{(j)} - \nabla\mathbf{K}_t(X_t^{(j)})^\top M_t^{-1} \sum_{k=1}^{J}\left(\frac{1}{J} - w_t^{(k)}\right)\mathbf{K}_t(X_t^{(k)}), \quad (17)$$

with $j \in \{1, \ldots, J\}$, $t \in [0, 1]$, and $\{X_0^{(j)}\}_{j=1}^J \overset{\text{i.i.d.}}{\sim} \pi_0$. Although (17) is distinct from (9) in discrete time, in *continuous* time the two interacting particle systems are equivalent:

Theorem 4.1. *In the limit $\Delta t \to 0$, Equation (17) approaches Equation (9). Thus the IPS obtained via sample-driven optimal transport (17) can be viewed as arising from mean-field model (8).*

The proof of this result follows from simple calculus and is contained in Appendix A.2. Owing to this equivalence in continuous time, we refer to the interacting particle system (17) as **KFRFlow-Importance (KFRFlow-I)**.

Theorem 4.1 highlights the fact that linearizing a Monge–Ampère equation for *static* optimal transport between π_t and $\pi_{t+\Delta t}$ results in a Poisson equation, and demonstrates that as $\Delta t \to 0$ this linearization yields the correct velocity field for the controlled *dynamic* minimum-energy transport problem of Section 3. Furthermore, it elucidates connections between SMC approaches based on tempered self-normalized importance sampling and Fisher–Rao gradient flows.

5. Implementation

KFRFlow (9) can be discretized in time, for example, via the explicit Euler method

$$X_{t+\Delta t}^{(j)} = X_t^{(j)} + \nabla\mathbf{K}_t(X_t^{(j)})^\top M_t^{-1} \cdot \frac{\Delta t}{J}\sum_{k=1}^{J}\left(\log\tfrac{\pi_1}{\pi_0}(X_t^{(k)}) - \frac{1}{J}\sum_{i=1}^{J}\log\tfrac{\pi_1}{\pi_0}(X_t^{(i)})\right)\mathbf{K}_t(X_t^{(k)}), \quad (18)$$

or any other standard ODE integration scheme, while KFRFlow-I (17) already has the form of a discrete-time iteration. In either case we start from $\{X_0^{(j)}\}_{j=1}^J \overset{\text{i.i.d.}}{\sim} \pi_0$ and simulate for unit time to obtain $\{X_1^{(j)}\}_{j=1}^J \sim \pi_{X_1} \approx \pi_1$. KFRFlow-I (17) and the Euler discretization of KFRFlow (18) are almost identical, and in fact (17) can be recovered, up to multiplication by a constant close to one, from (18) by applying the approximation $\Delta t \log y = \log y^{\Delta t} \approx y^{\Delta t} - 1$ to all instances of $\Delta t \log \frac{\pi_1}{\pi_0}$ in (18).

In practice, the performances of KFRFlow-I (17) and discretizations of KFRFlow (9) are often similar, but we have noticed that for large Δt or particularly challenging sampling tasks, such as those in high dimensions, KFRFlow-I tends to be more stable. One possible explanation for this advantage is that in KFRFlow-I we evaluate $(\frac{\pi_1}{\pi_0})^{\Delta t}$ rather than $\Delta t \log \frac{\pi_1}{\pi_0}$, which is a more stable computation when the ratio $\frac{\pi_1}{\pi_0}$ is small. Another consideration is that the update in KFRFlow-I (17) is derived as an approximate transport *map* between π_t and $\pi_{t+\Delta t}$, whereas KFRFlow updates as in (18) are discretizations of a continuous flow (9). These discretizations may not be good transport maps, especially for large Δt, in which case KFRFlow-I, which is explicitly designed for discrete-time transport, may perform better.

5.1. Numerical Stability

Choice of KFRFlow versus KFRFlow-I aside, in our experiments we have noticed that the matrix M_t in KFRFlow (9) and KFRFlow-I (17) may at times be poorly conditioned, leading to numerical instability or poor quality samples. We have found two helpful tactics for mitigating this issue: inflating the diagonal of M_t and introducing noise.

5.1.1. Regularization of M_t

Issues of ill-conditioning of M_t can be ameliorated by replacing M_t in Equations (17) and (18) with $M_{t,\lambda} = M_t + \lambda I$ for some $\lambda > 0$; this is essentially a Tikhonov regularization of (6) or (16). Inflating the diagonal of M_t does not require additional information about π_1 and π_0 but does require finding an appropriate λ and potentially alters the time-dependent distribution $\rho_t = \mathrm{Law}(X_t)$.

5.1.2. Stochastic Modification

As noted similarly in, e.g., Song et al. (2021); Albergo et al. (2023), the continuity equation (3) can be written equivalently for any $\epsilon > 0$ as a Fokker–Planck equation

$$\partial_t \pi_t = -\nabla \cdot (\pi_t (v_t + \epsilon \nabla \log \pi_t)) + \epsilon \nabla^2 \pi_t, \tag{19}$$

by making use of the identity $\nabla \log \pi_t = \frac{\nabla \pi_t}{\pi_t}$ (we use ∇^2 to denote the Laplacian). The Fokker–Planck equation (19) corresponds to an SDE for X_t,

$$\mathrm{d}X_t = (v_t(X_t) + \epsilon \nabla \log \pi_t(X_t))\,\mathrm{d}t + \sqrt{2\epsilon}\,\mathrm{d}W_t, \quad t \in [0,1], \tag{20}$$

which possesses the same marginal distributions as the ODE (2). The SDE (20) can hence be used with the velocity field v_t (7) as the basis for a *stochastic* interacting particle system, which we refer to as Kernel Fisher–Rao Diffusion (KFRD), for traversing the geometric mixture π_t; see Appendix B for further details. Simulation of (20) does require access to the score of π_t and hence KFRD is not gradient-free, but in the case that gradients of π_0 and π_1 are available, the score of π_t is simply

$$\nabla \log \pi_t = (1-t)\nabla \log \pi_0 + t \nabla \log \pi_1.$$

We find that the introduction of noise through (20) often increases numerical stability and enhances sample quality, owing to the incorporation of gradient information.

5.2. Computational Cost

The naive complexity of computing the right-hand side of the KFRFlow ODE (9) or one step of KFRFlow-I (17) is $\mathcal{O}(J^3)$, as we require a solve with a $J \times J$ symmetric matrix. While symmetric linear solves are well-optimized computations, the cost of KFRFlow could be lowered in practice, for instance, by use of random features (Rahimi & Recht, 2007) or other kernel dimension-reduction techniques. We only demonstrate the "vanilla" $\mathcal{O}(J^3)$ implementation of KFRFlow in this work for clarity of presentation; more sophisticated implementation strategies, including use of random features, are part of ongoing work.

6. Numerical Examples

Now we present proof-of-concept examples demonstrating the efficacy of KFRFlow, KFRFlow-I, and KFRD in generating samples from various target distributions. We also compare the performances of our algorithms to those of ensemble Kalman inversion stopped at $t = 1$ (EKI, Iglesias et al. (2013)), the ensemble Kalman sampler (EKS, Garbuno-Inigo et al. (2020a)), consensus-based sampling (CBS, Carrillo et al. (2022)), Stein variational gradient descent (SVGD, Liu & Wang (2016)), and the unadjusted Langevin algorithm (ULA, e.g., Roberts & Tweedie (1996)). Of the competing algorithms, EKI, EKS, and CBS are gradient-free, and hence we employ them as bases for comparison for KFRFlow and KFRFlow-I, while SVGD and ULA require $\nabla \log \pi_1$, so we compare their performances to those of KFRD. Like KFRFlow(-I) and KFRD, EKI, EKS, CBS, and SVGD are interacting particle systems which evolve ensembles of J particles together such that their collective distribution approaches π_1. By contrast, ULA is a Markov chain algorithm which does not make use of interaction, but to obtain a similarly structured sampler to the IPS algorithms we use ULA in "parallel mode," simulating J independent chains initialized at points randomly drawn from $\mathcal{N}(0, I_d)$ and retaining the final state of each chain to form a set of J samples from the target.

In our experiments we take the kernel in KFRFlow(-I), KFRD, and SVGD to be inverse multiquadric (IMQ)

$$K(x,x') = \left(1 + \frac{\|x-x'\|^2}{h^2}\right)^{-1/2} \tag{21}$$

with bandwidth $h > 0$ selected at each step of the iterations according to the median heuristic (Liu & Wang, 2016). We assess sample quality using kernel Stein discrepancy (KSD) (Gorham & Mackey, 2020) with the IMQ kernel (21) with bandwidth $h = 1$. The reference distribution π_0 is always standard Gaussian. We perform all experiments in Julia using the package `DifferentialEquations.jl` (Rackauckas & Nie, 2017) to integrate the ODEs and SDEs associated with KFRFlow, KFRD, EKI, EKS, CBS, SVGD, and ULA. Code for the experiments is available at at `https://github.com/amaurais/KFRFlow.jl`.

6.1. Two-Dimensional Bayesian Posteriors

We apply KFRFlow (9) and KFRFlow-I (17) to sample three two-dimensional densities. In all three cases π_1 is a Bayesian posterior proportional to $\pi_0 \ell$ for a likelihood of the form $\ell(x) \propto \exp\left(-\frac{1}{\sigma_\varepsilon^2}\|y^* - G(x)\|_2^2\right)$, i.e., $y^* \in \mathbb{R}$ is Gaussian with mean $G(x)$ and variance σ_ε^2. Definitions of the three likelihoods may be found in Appendix C.1.

Figure 2 displays $J = 300$ samples obtained from a forward Euler discretization of KFRFlow with uniform timestep $\Delta t = 0.01$. The samples at $t = 1$ are qualitatively consistent with the target densities for each example.

In Figure 3 we compare the performances of KFRFlow and KFRFlow-I to those of fellow gradient-free sampling

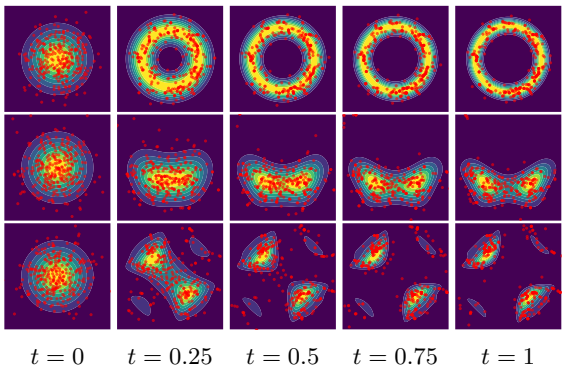

Figure 2. **Two-dimensional posteriors:** samples at $t \in \{0, 0.25, 0.5, 0.75, 1\}$ generated by KFRFlow (9) for the donut (top), butterfly (middle), and spaceships (bottom) examples.

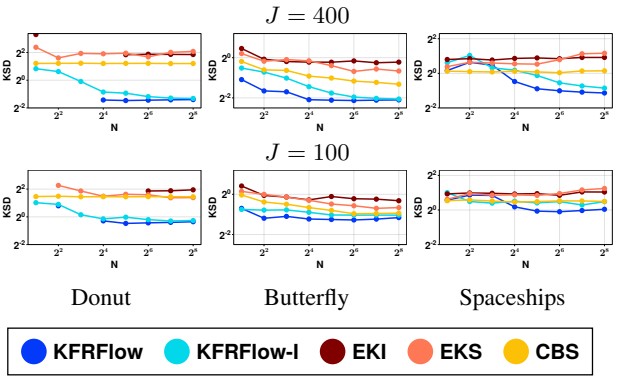

Figure 3. **Two-dimensional posteriors:** average KSD at stopping time between π_1 and ensembles of size $J \in \{100, 400\}$ generated by gradient-free samplers. A missing point indicates that a method was unstable at that setting of N.

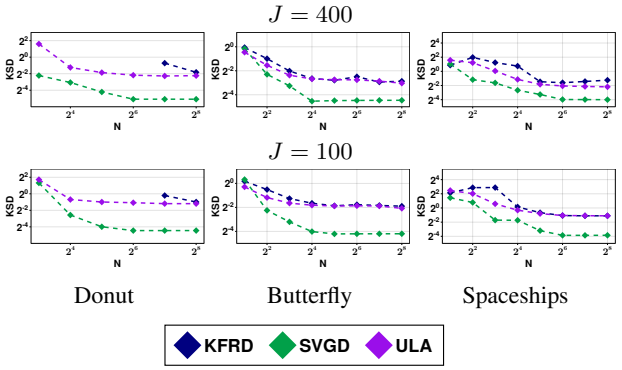

Figure 4. **Two-dimensional posteriors:** average KSD at stopping time between π_1 and ensembles of size $J \in \{100, 400\}$ generated by gradient-based samplers. A missing point indicates that a method was unstable at that setting of N.

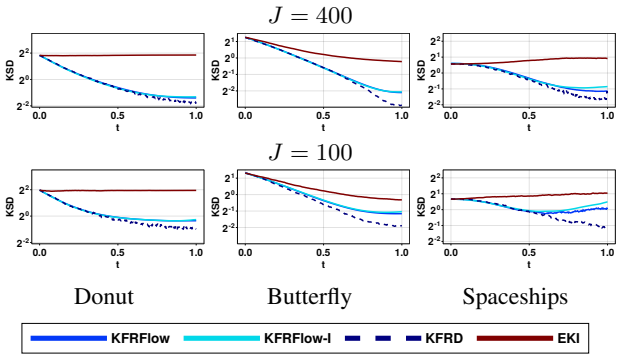

Figure 5. **Two-dimensional posteriors:** evolution of KSD with t for the unit-time methods KFRFlow, KFRFlow-I, KFRD, and EKI for ensembles of $J = 400$ and $J = 100$ with $\Delta t = 2^{-8}$. KFRD is plotted with dashed lines because it requires gradients, whereas KFRFlow(-I) and EKI are gradient-free.

algorithms EKI, EKS, and CBS, and in Figure 4 we compare the performances of gradient-based KFRD, SVGD, and ULA. We use the algorithms to generate target ensembles of size $J \in \{25, 50, 100, 200, 400\}$ with number of steps $N \in \{2^1, 2^2, \ldots, 2^8\}$. For unit-time KFRFlow(-I), KFRD, and EKI, the resulting step-size is $1/N$, but for infinite-time EKS, CBS, SVGD, and ULA we must choose a stopping time $T > 0$, resulting in a step-size of T/N. We test a range of stopping times T for EKS, CBS, SVGD, and ULA, regularization levels λ for KFRFlow(-I), noise levels ϵ for KFRD, and temperature β for CBS and report KSD corresponding to the best parameter settings for each algorithm and each (J, N). We use a fourth-order Adams–Bashforth discretization of KFRFlow because we find that it generates better samples than forward Euler at little additional cost, while by contrast we use forward Euler for SVGD because we find that SVGD does not benefit from multistep discretizations in these examples. We use Euler–Maruyama discretizations for KFRD, EKI, EKS, CBS, and ULA. The values of KSD we report are averages over 30 trials.

In Figure 3 we see that KFRFlow and KFRFlow-I generally produce better-quality samples, as measured with KSD, than EKI, EKS, and CBS. There are some settings in the donut example at which KFRFlow or EKI is unstable, but, interestingly, KFRFlow-I is stable across all settings of (J, N), even in some cases when the gradient-based algorithms KFRD and SVGD (Figure 4) are not. In Figure 4 we see that the performance of KFRD is comparable to that of ULA and generally exceeded by SVGD in these examples, though for small N it is often the case that KFRFlow or KFRFlow-I yields comparable or better performance than SVGD. For additional details and results see Appendix C.1.

6.2. Higher-Dimensional Funnel Distributions

Here for dimension $d \in \{5, 10, 15, 20\}$ we compare the performance of KFRFlow-I, KFRD, CBS, SVGD, and

ULA in sampling from "funnel" distributions of the form $\pi_1(\mathbf{x}) = \mathcal{N}(x_1; 0, 9)\mathcal{N}(\mathbf{x}_{2:d}; \mathbf{0}, \exp(x_1)\mathbf{I})$. This family of distributions appears in Neal (2003) and is a common benchmark for sampling algorithms, e.g., Arbel et al. (2021); Zhang et al. (2023); Xu et al. (2023b).

For each setting of d we apply these algorithms to generate $J = 100$ samples from π_1. For KFRFlow-I and KFRD we set $\Delta t = 0.01$, corresponding to $N = 100$ steps for the infinite-time algorithms CBS, SVGD, and ULA. As in Section 6.1 we optimize the hyperparameters for KFRFlow-I, KFRD, CBS, SVGD, and ULA via coarse direct search to minimize KSD between the final samples and π_1. EKI and EKS are not applicable to the funnel because it is not a Bayesian posterior with a Gaussian likelihood, and we focus on KFRFlow-I rather than KFRFlow due to its demonstrated stability. The data in Figure 6 and Figure 7 are reflective of averaging the results of 30 independent trials.

In Figure 6 we plot KSD between the ensembles and the funnel targets as a function of dimension d. We see that KSD increases with dimension for the gradient-based algorithms and that KFRD is competitive with SVGD and ULA at all values of d. We also see that KFRFlow-I generates better quality samples than CBS and, interestingly, that the quality of the samples generated by both of these gradient-free IPS algorithms does not seem to be meaningfully impacted by dimension. A more thorough investigation of this phenomenon, also visible in Figure 7, is a topic for future work. For additional details and results see Appendix C.2.

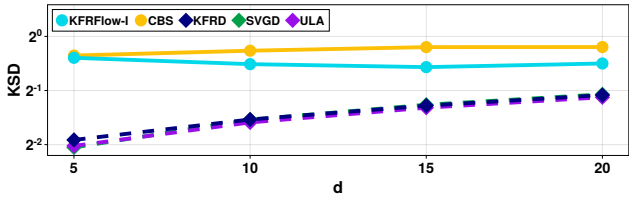

Figure 6. **Funnels:** average KSD at stopping time between π_1 and samples generated by KFRFlow-I, KFRD, CBS, SVGD, and ULA for $d \in \{5, 10, 15, 20\}$. Gradient-free algorithms are plotted with solid lines, while gradient-based algorithms are plotted with dashed lines.

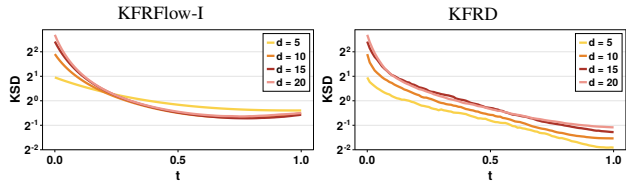

Figure 7. **Funnels:** evolution of KSD between π_1 and samples at time t for KFRFlow-I (left) and KFRD (right).

7. Discussion and Future Work

We have introduced a mean-field ODE and corresponding interacting particle systems which approximately transport samples from π_0 to π_1 in unit time. We obtain the mean-field ODE by solving an elliptic PDE arising from the Fisher–Rao gradient flow of the negative log likelihood under the ansatz that the solution lies in a reproducing kernel Hilbert space. The RKHS form of the mean-field ODE gives rise to tractable, gradient-free interacting particle systems for sampling.

Several lines of inquiry would enhance our understanding of the mean-field model and KFRFlow interacting particle systems. The tempered likelihood path $\pi_t \propto \pi_0^{1-t}\pi_1^t$ employed here is itself an interesting object of study: in addition to being a segment of the Fisher-Rao gradient flow of $\mathcal{F}(\mu) = -\mathbb{E}_\mu[\log \frac{\pi_1}{\pi_0}]$, $(\pi_t)_{t\in[0,1]}$ can also be characterized as a unit-time rescaling of a Fisher-Rao gradient flow of $\mu \mapsto D_{\mathrm{KL}}(\mu\|\pi_1)$ (Domingo-Enrich & Pooladian, 2023) and as a path of Kullback–Leibler divergence barycenters (e.g., Amari (2016, Theorem 4.9)); that is,

$$\pi_t = \arg\min_{\mu} (1-t)D_{\mathrm{KL}}(\mu\|\pi_0) + tD_{\mathrm{KL}}(\mu\|\pi_1), \quad t \in [0,1].$$

Thus, KFRFlow can be equally well viewed as (i) early-stopping of maximum likelihood, (ii) gradient descent of $D_{\mathrm{KL}}(\cdot\|\pi_1)$, or (iii) a continuation method for minimizing $D_{\mathrm{KL}}(\cdot\|\pi_1)$. Perhaps equally intriguing is that to produce the KFRFlow mean-field model (8), we take the path $(\pi_t)_{t\in[0,1]}$ natively corresponding to Fisher-Rao gradient flow and "Wasserstein-ize" it: we seek potentials $u_t : \mathbb{R}^d \to \mathbb{R}$ such that $-\nabla \cdot (\pi_t \nabla u_t) = \partial_t \pi_t$, leading to particle algorithms that transport samples collectively along $(\pi_t)_{t\in[0,1]}$ via the action of a gradient velocity field. Understanding how to interpret the path $(\pi_t)_{t\in[0,1]}$ through a (kernelized) Wasserstein lens is thus of great interest.

On a less abstract level, KFRFlow has appealing properties for practical scientific sampling applications—it is gradient-free, closed-form, inherently finite-time, and only requires an unnormalized likelihood—but focused effort is needed to move from "vanilla" algorithms as in (17) and (18) to more sophisticated implementations suitable for challenging problems in high dimensions. We are currently investigating computational strategies for this purpose, including reducing computational complexity through kernel appproximations and dimension reduction, exploiting or imposing structure and sparsity through well-designed choices of kernel, and adaptive time-stepping for the resulting systems of ODEs. Finally, strengthening the numerical and statistical analysis of KFRFlow by understanding questions of approximation error and sample complexity, particularly as these quantities relate to choice of kernel and time-stepping schedule, is a closely related and important area for future work, which will inform the development of KFRFlow generalizations.

Acknowledgements

The authors thank the anonymous reviewers for their helpful suggestions and discussion. AM and YM were supported by the Office of Naval Research, SIMDA (Sea Ice Modeling and Data Assimilation) MURI, award number N00014-20-1-2595 (Dr. Reza Malek-Madani and Dr. Scott Harper). AM was additionally supported by the NSF Graduate Research Fellowship under Grant No. 1745302.

Impact Statement

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions. (arXiv:2303.08797), March 2023.

Amari, S.-i. *Information geometry and its applications*, volume 194. Springer, 2016.

Arbel, M., Matthews, A., and Doucet, A. Annealed flow transport monte carlo. In *International Conference on Machine Learning*, pp. 318–330. PMLR, 2021.

Baptista, R., Hosseini, B., Kovachki, N. B., and Marzouk, Y. Conditional Sampling with Monotone GANs: From Generative Models to Likelihood-Free Inference. (arXiv:2006.06755), June 2023a. doi: 10.48550/arXiv.2006.06755.

Baptista, R., Marzouk, Y., and Zahm, O. On the representation and learning of monotone triangular transport maps. *Foundations of Computational Mathematics*, in press, 2023b. arXiv:2009.10303.

Brekelmans, R., Masrani, V., Bui, T., Wood, F., Galstyan, A., Steeg, G. V., and Nielsen, F. Annealed Importance Sampling with q-Paths. (arXiv:2012.07823), December 2020. doi: 10.48550/arXiv.2012.07823.

Brenier, Y. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991.

Brennan, M. C., Bigoni, D., Zahm, O., Spantini, A., and Marzouk, Y. Greedy inference with structure-exploiting lazy maps. *Advances in Neural Information Processing Systems*, 33:8330–8342, 2020.

Bunne, C., Krause, A., and Cuturi, M. Supervised training of conditional Monge maps. *Advances in Neural Information Processing Systems*, 35:6859–6872, 2022.

Carrillo, J. A., Hoffmann, F., Stuart, A. M., and Vaes, U. Consensus-based sampling. *Studies in Applied Mathematics*, 148(3):1069–1140, 2022. ISSN 1467-9590. doi: 10.1111/sapm.12470.

Chen, J. and Revels, J. Robust benchmarking in noisy environments. *arXiv preprint arXiv:1608.04295*, 2016.

Chen, P., Wu, K., Chen, J., O'Leary-Roseberry, T., and Ghattas, O. Projected Stein variational Newton: A fast and scalable Bayesian inference method in high dimensions. *Advances in Neural Information Processing Systems*, 32, 2019.

Chen, Y., Huang, D. Z., Huang, J., Reich, S., and Stuart, A. M. Gradient Flows for Sampling: Mean-Field Models, Gaussian Approximations and Affine Invariance, July 2023.

Chewi, S. *Log-Concave Sampling*. In Progress, 2023.

Dai, B. and Seljak, U. Sliced Iterative Normalizing Flows. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 2352–2364. PMLR, July 2021.

Daum, F. and Huang, J. Particle flow for nonlinear filters. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5920–5923. IEEE, 2011.

Daum, F. and Huang, J. Particle flow for nonlinear filters, Bayesian decisions and transport. In *Proceedings of the 16th International Conference on Information Fusion*, pp. 1072–1079. IEEE, 2013.

De Bortoli, V., Thornton, J., Heng, J., and Doucet, A. Diffusion Schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17695–17709. Curran Associates, Inc., 2021.

Del Moral, P., Doucet, A., and Jasra, A. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436, 2006.

Dia, B. M. A Continuation Method in Bayesian Inference. *SIAM/ASA Journal on Uncertainty Quantification*, pp. 646–681, June 2023. doi: 10.1137/19M130251X.

Ding, Z. and Li, Q. Ensemble Kalman inversion: Mean-field limit and convergence analysis. *Statistics and Computing*, 31(1):9, January 2021. ISSN 1573-1375. doi: 10.1007/s11222-020-09976-0.

Domingo-Enrich, C. and Pooladian, A.-A. An Explicit Expansion of the Kullback-Leibler Divergence along its Fisher-Rao Gradient Flow. *Transactions on Machine Learning Research*, March 2023. ISSN 2835-8856.

Earl, D. J. and Deem, M. W. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916, 2005.

Evans, L. C. Partial differential equations and Monge-Kantorovich mass transfer. *Current developments in mathematics*, 1997(1):65–126, 1997.

Garbuno-Inigo, A., Hoffmann, F., Li, W., and Stuart, A. M. Interacting Langevin Diffusions: Gradient Structure and Ensemble Kalman Sampler. *SIAM Journal on Applied Dynamical Systems*, 19(1):412–441, January 2020a. doi: 10.1137/19M1251655.

Garbuno-Inigo, A., Nüsken, N., and Reich, S. Affine Invariant Interacting Langevin Dynamics for Bayesian Inference. *SIAM Journal on Applied Dynamical Systems*, July 2020b. doi: 10.1137/19M1304891.

Geyer, C. J. Markov chain Monte Carlo maximum likelihood. 1991.

Gorham, J. and Mackey, L. Measuring Sample Quality with Kernels, October 2020.

Goshtasbpour, S., Cohen, V., and Perez-Cruz, F. Adaptive annealed importance sampling with constant rate progress. In *International Conference on Machine Learning*, pp. 11642–11658. PMLR, 2023.

Heng, J., De Bortoli, V., and Doucet, A. Diffusion schrödinger bridges for bayesian computation. *Statistical Science*, 39(1):90–99, 2024.

Iglesias, M. A., Law, K. J. H., and Stuart, A. M. Ensemble Kalman methods for inverse problems. *Inverse Problems*, 29(4):045001, April 2013. ISSN 0266-5611, 1361-6420. doi: 10.1088/0266-5611/29/4/045001.

Jaini, P., Selby, K. A., and Yu, Y. Sum-of-squares polynomial flow. In *International Conference on Machine Learning*, pp. 3009–3018. PMLR, 2019.

Katzfuss, M. and Schäfer, F. Scalable Bayesian Transport Maps for High-Dimensional Non-Gaussian Spatial Fields. *Journal of the American Statistical Association*, 0(0):1–15, 2023. ISSN 0162-1459. doi: 10.1080/01621459.2023.2197158.

Kobyzev, I., Prince, S. J., and Brubaker, M. A. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979, 2020.

Korba, A. and Portier, F. Adaptive importance sampling meets mirror descent: a bias-variance tradeoff. In *International Conference on Artificial Intelligence and Statistics*, pp. 11503–11527. PMLR, 2022.

Kuang, M. and Tabak, E. G. Sample-Based Optimal Transport and Barycenter Problems. *Communications on Pure and Applied Mathematics*, 72(8):1581–1630, 2019. ISSN 1097-0312. doi: 10.1002/cpa.21848.

Künsch, H. R. Recursive Monte Carlo filters: Algorithms and theoretical analysis. *The Annals of Statistics*, 33(5):1983 – 2021, 2005. doi: 10.1214/009053605000000426. URL https://doi.org/10.1214/009053605000000426.

Laugesen, R. S., Mehta, P. G., Meyn, S. P., and Raginsky, M. Poisson's equation in nonlinear filtering. *SIAM Journal on Control and Optimization*, 53(1):501–525, 2015.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. (arXiv:2210.02747), 2023. doi: 10.48550/arXiv.2210.02747. URL http://arxiv.org/abs/2210.02747.

Liu, Q. and Wang, D. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. (arXiv:2209.03003), 2022. doi: 10.48550/arXiv.2209.03003. URL http://arxiv.org/abs/2209.03003.

Marzouk, Y., Moselhy, T., Parno, M., and Spantini, A. Sampling via measure transport: An introduction. *Handbook of uncertainty quantification*, 1:2, 2016.

Myers, A., Thiéry, A. H., Wang, K., and Bui-Thanh, T. Sequential ensemble transform for Bayesian inverse problems. *Journal of Computational Physics*, 427:110055, February 2021. ISSN 0021-9991. doi: 10.1016/j.jcp.2020.110055.

Neal, R. M. Annealed importance sampling. *Statistics and computing*, 11:125–139, 2001.

Neal, R. M. Slice sampling. *The annals of statistics*, 31(3):705–767, 2003.

Owen, A. B. *Monte Carlo theory, methods and examples*. https://artowen.su.domains/mc/, 2013.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

Rackauckas, C. and Nie, Q. DifferentialEquations.jl–a performant and feature-rich ecosystem for solving differential equations in Julia. *Journal of Open Research Software*, 5(1), 2017.

Rahimi, A. and Recht, B. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

Reich, S. A dynamical systems framework for intermittent data assimilation. *BIT Numerical Mathematics*, 51(1): 235–249, March 2011. ISSN 1572-9125. doi: 10.1007/s10543-010-0302-4.

Reich, S. A Nonparametric Ensemble Transform Method for Bayesian Inference. *SIAM Journal on Scientific Computing*, 35(4):A2013–A2024, January 2013. ISSN 1064-8275, 1095-7197. doi: 10.1137/130907367. URL http://epubs.siam.org/doi/10.1137/130907367.

Reich, S. and Cotter, C. *Probabilistic forecasting and Bayesian data assimilation*. Cambridge University Press, 2015.

Reich, S. and Weissmann, S. Fokker–Planck Particle Systems for Bayesian Inference: Computational Approaches. *SIAM/ASA Journal on Uncertainty Quantification*, 9(2):446–482, January 2021. ISSN 2166-2525. doi: 10.1137/19M1303162.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

Robert, C. P. and Casella, G. The Metropolis—Hastings algorithm. In *Monte Carlo Statistical Methods*, pp. 267–320. Springer New York, New York, NY, 2004. ISBN 978-1-4757-4145-2. doi: 10.1007/978-1-4757-4145-2_7.

Roberts, G. O. and Tweedie, R. L. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, pp. 341–363, 1996.

Ruchi, S., Dubinkina, S., and Iglesias, M. A. Transform-based particle filtering for elliptic Bayesian inverse problems. *Inverse Problems*, 35(11):115005, October 2019. ISSN 0266-5611. doi: 10.1088/1361-6420/ab30f3.

Ruchi, S., Dubinkina, S., and de Wiljes, J. Fast hybrid tempered ensemble transform filter formulation for Bayesian elliptical problems via Sinkhorn approximation. *Nonlinear Processes in Geophysics*, 28(1):23–41, January 2021. ISSN 1023-5809. doi: 10.5194/npg-28-23-2021.

Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, 2008.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International conference on learning representations*, 2021.

Spantini, A., Bigoni, D., and Marzouk, Y. Inference via low-dimensional couplings. *The Journal of Machine Learning Research*, 19(1):2639–2709, 2018.

Spantini, A., Baptista, R., and Marzouk, Y. Coupling Techniques for Nonlinear Ensemble Filtering. *SIAM Review*, 64(4):921–953, November 2022. ISSN 0036-1445, 1095-7200. doi: 10.1137/20M1312204.

Steinwart, I. and Christmann, A. *Kernels and Reproducing Kernel Hilbert Spaces*, pp. 110–163. Springer New York, New York, NY, 2008. ISBN 978-0-387-77242-4. doi: 10.1007/978-0-387-77242-4_4. URL https://doi.org/10.1007/978-0-387-77242-4_4.

Syed, S., Romaniello, V., Campbell, T., and Bouchard-Côté, A. Parallel tempering on optimized paths. In *International Conference on Machine Learning*, pp. 10033–10042. PMLR, 2021.

Taghvaei, A. and Hosseini, B. An Optimal Transport Formulation of Bayes' Law for Nonlinear Filtering Algorithms. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 6608–6613, December 2022. doi: 10.1109/CDC51059.2022.9992776.

Taghvaei, A. and Mehta, P. G. A survey of feedback particle filter and related controlled interacting particle systems (CIPS). *Annual Reviews in Control*, 55:356–378, January 2023. ISSN 1367-5788. doi: 10.1016/j.arcontrol.2023.03.006.

Tian, Y., Panda, N., and Lin, Y. T. Liouville Flow Importance Sampler. In *Proceedings of the 41st International Conference on Machine Learning*. PMLR, July 2024. doi: 10.48550/arXiv.2405.06672.

Trigila, G. and Tabak, E. G. Data-Driven Optimal Transport. *Communications on Pure and Applied Mathematics*, 69 (4):613–648, 2016. ISSN 1097-0312. doi: 10.1002/cpa.21588.

Trillos, N. G., Hosseini, B., and Sanz-Alonso, D. From optimization to sampling through gradient flows. *Notices of the American Mathematical Society*, 70(6), 2023.

Vargas, F., Grathwohl, W., and Doucet, A. Denoising diffusion samplers. *arXiv preprint arXiv:2302.13834*, 2023a.

Vargas, F., Padhy, S., Blessing, D., and Nüsken, N. Transport meets Variational Inference: Controlled Monte Carlo Diffusions. In *The Twelfth International Conference on Learning Representations*, October 2023b.

Villani, C. *Topics in Optimal Transportation*. American Mathematical Soc., August 2021. ISBN 978-1-4704-6726-5.

Wang, L. and Nüsken, N. Measure transport with kernel mean embeddings, January 2024.

Xu, C., Cheng, X., and Xie, Y. Optimal transport flow and infinitesimal density ratio estimation. (arXiv:2305.11857), 2023a. doi: 10.48550/arXiv.2305.11857. URL http://arxiv.org/abs/2305.11857.

Xu, Z., Chen, N., and Campbell, T. Mixflows: principled variational inference via mixed flows. In *International Conference on Machine Learning*, pp. 38342–38376. PMLR, 2023b.

Yang, J., Roberts, G. O., and Rosenthal, J. S. Optimal scaling of random-walk metropolis algorithms on general target distributions. *Stochastic Processes and their Applications*, 130(10):6094–6132, 2020.

Zhang, D., Chen, R. T. Q., Liu, C.-H., Courville, A., and Bengio, Y. Diffusion Generative Flow Samplers: Improving learning signals through partial trajectory optimization, October 2023.

A. Derivations and Proofs

A.1. Derivation of Kernel Fisher-Rao Flow Interacting Particle System

We would like to approximate the mean-field ODE (8)

$$\dot{X}_t = v_t(X_t) = \mathbb{E}_{\rho_t}\left[\nabla_1 K(X_t, X')M_{\rho_t}^{-1}K_{\rho_t}\left(\log\tfrac{\pi_1}{\pi_0} - \mathbb{E}_{\rho_t}\left[\log\tfrac{\pi_1}{\pi_0}\right]\right)(X')\right], \tag{22}$$

with an interacting particle system $\{X_t^{(j)}\}_{j=1}^J$. To obtain such an IPS, we approximate the expectations in (22) via Monte Carlo. Beginning with the outer expectation, we have

$$\dot{X}_t^{(j)} \approx \frac{1}{J}\sum_{n=1}^{J}\nabla_1 K(X_t^{(j)}, X_t^{(n)})M_{\rho_t}^{-1}K_{\rho_t}\left(\log\tfrac{\pi_1}{\pi_0} - \mathbb{E}_{\rho_t}\left[\log\tfrac{\pi_1}{\pi_0}\right]\right)(X_t^{(n)}). \tag{23}$$

Next, we obtain a Monte Carlo approximation of $f_t \equiv M_{\rho_t}^{-1}K_{\rho_t}\left(\log\tfrac{\pi_1}{\pi_0} - \mathbb{E}_{\rho_t}\left[\log\tfrac{\pi_1}{\pi_0}\right]\right)$ by examining the system $M_{\rho_t}f_t = K_{\rho_t}\left(\log\tfrac{\pi_1}{\pi_0} - \mathbb{E}_{\rho_t}\left[\log\tfrac{\pi_1}{\pi_0}\right]\right)$,

$$\iint\limits_{\mathbb{R}^d\times\mathbb{R}^d} f_t(z)\,\langle\nabla_1 K(y,\cdot),\ \nabla_1 K(y,z)\rangle\,\mathrm{d}\pi_t(y)\,\mathrm{d}\pi_t(z) = \int_{\mathbb{R}^d} K(\cdot,y)\left(\log\tfrac{\pi_1}{\pi_0}(y) - \mathbb{E}_{\pi_t}\left[\log\tfrac{\pi_1}{\pi_0}\right]\right)\mathrm{d}\pi_t(y).$$

and approximating expectations on both sides via Monte Carlo,

$$\frac{1}{J^2}\sum_{m=1}^{J}\sum_{i=1}^{J} f_t(X_t^{(m)})\left\langle\nabla_1 K(X_t^{(i)},\cdot),\ \nabla_1 K(X_t^{(i)},X_t^{(m)})\right\rangle = \frac{1}{J}\sum_{k=1}^{J}K(\cdot,X_t^{(k)})\left(\log\tfrac{\pi_1}{\pi_0}(X_t^{(k)}) - \frac{1}{J}\sum_{i=1}^{J}\log\tfrac{\pi_1}{\pi_0}(X_t^{(i)})\right). \tag{24}$$

We enforce (24) at $\{X_t^{(\ell)}\}_{\ell=1}^J$ in order to obtain a system of J equations for $f_t(X_t^{(1)}),\ldots,f_t(X_t^{(J)})$,

$$\begin{aligned}\frac{1}{J^2}\sum_{m=1}^{J}\sum_{i=1}^{J} f_t(X_t^{(m)})&\left\langle\nabla_1 K(X_t^{(i)},X_t^{(\ell)}),\ \nabla_1 K(X_t^{(i)},X_t^{(m)})\right\rangle \\ &= \frac{1}{J}\sum_{k=1}^{J}K(X_t^{(\ell)},X_t^{(k)})\left(\log\tfrac{\pi_1}{\pi_0}(X_t^{(k)}) - \frac{1}{J}\sum_{i=1}^{J}\log\tfrac{\pi_1}{\pi_0}(X_t^{(i)})\right), \quad \ell = 1,\ldots,J,\end{aligned} \tag{25}$$

which we write succinctly as

$$M_t\mathbf{f}_t = \sum_{k=1}^{J}\mathbf{K}_t(X_t^{(k)})\left(\log\tfrac{\pi_1}{\pi_0}(X_t^{(k)}) - \frac{1}{J}\sum_{i=1}^{J}\log\tfrac{\pi_1}{\pi_0}(X_t^{(i)})\right), \tag{26}$$

where $\mathbf{f}_t = (f_t(X_t^{(1)}),\ldots,f_t(X_t^{(J)}))$, $\mathbf{K}_t(\cdot) = (K(\cdot,X_t^{(1)}),\ldots,K(\cdot,X_t^{(J)}))$, and $M_t \in \mathbb{R}^{J\times J}$ is given by

$$(M_t)_{\ell,m} = \frac{1}{J}\sum_{i=1}^{J}\langle\nabla_1 K(X_t^{(i)},X_t^{(\ell)}),\ \nabla_1 K(X_t^{(i)},X_t^{(m)})\rangle, \quad \ell,m = 1,\ldots,J.$$

Notice that to arrive at (26) we have canceled a common factor of $\frac{1}{J}$ on either side of (25). Hence, we have

$$\mathbf{f}_t = M_t^{-1}\sum_{k=1}^{J}\mathbf{K}_t(X_t^{(k)})\left(\log\tfrac{\pi_1}{\pi_0}(X_t^{(k)}) - \frac{1}{J}\sum_{i=1}^{J}\log\tfrac{\pi_1}{\pi_0}(X_t^{(i)})\right),$$

and we approximate $M_{\rho_t}^{-1}K_{\rho_t}\left(\log\tfrac{\pi_1}{\pi_0} - \mathbb{E}_{\rho_t}\left[\log\tfrac{\pi_1}{\pi_0}\right]\right)(X_t^{(\ell)})$ in (23) with $f_t(X_t^{(\ell)})$, writing the result in vector form,

$$\dot{X}_t^{(j)} = \tfrac{1}{J}\left(\nabla_1 K(X_t^{(j)},X_t^{(1)}),\ldots,\nabla_1 K(X_t^{(j)},X_t^{(J)})\right)M_t^{-1}\sum_{k=1}^{J}\mathbf{K}_t(X_t^{(k)})\left(\log\tfrac{\pi_1}{\pi_0}(X_t^{(k)}) - \frac{1}{J}\sum_{i=1}^{J}\log\tfrac{\pi_1}{\pi_0}(X_t^{(i)})\right),$$

and obtaining the IPS (9).

A.2. Proof of Theorem 4.1

Notice that time only enters the update equation (17) through the importance weights $w_t^{(k)}$. To obtain the continuous time limiting ODE we rearrange, divide by Δt on both sides, and take $\Delta t \to 0$,

$$\lim_{\Delta t\to 0} \frac{X_{t+\Delta t}^{(j)} - X_t^{(j)}}{\Delta t} = \lim_{\Delta t\to 0} -\begin{pmatrix}\nabla_1 K(X_t^{(j)}, X_t^{(1)}) & \cdots & \nabla_1 K(X_t^{(j)}, X_t^{(J)})\end{pmatrix} M_t^{-1} \sum_{k=1}^J \frac{\frac{1}{J} - w_t^{(k)}}{\Delta t} \begin{pmatrix} K(X_t^{(k)}, X_t^{(1)}) \\ \vdots \\ K(X_t^{(k)}, X_t^{(J)})\end{pmatrix}.$$

Examining the terms above involving Δt, we see that for $k \in \{1, \dots, J\}$ we have

$$\begin{aligned}
\lim_{\Delta t\to 0} \frac{\frac{1}{J} - w_t^{(k)}}{\Delta t} &= -\lim_{\Delta t\to 0} \frac{\frac{(\frac{\pi_1}{\pi_0}(X_t^{(k)}))^{\Delta t}}{\sum_{i=1}^J (\frac{\pi_1}{\pi_0}(X_t^{(i)}))^{\Delta t}} - \frac{1}{J}}{\Delta t} = -\lim_{\Delta t\to 0} \frac{\frac{(\frac{\pi_1}{\pi_0}(X_t^{(k)}))^{\Delta t}}{\sum_{i=1}^J (\frac{\pi_1}{\pi_0}(X_t^{(i)}))^{\Delta t}} - \frac{(\frac{\pi_1}{\pi_0}(X_t^{(k)}))^{0}}{\sum_{i=1}^J (\frac{\pi_1}{\pi_0}(X_t^{(i)}))^{0}}}{\Delta t} \\
&= -\frac{\mathrm{d}}{\mathrm{d}s} \frac{(\frac{\pi_1}{\pi_0}(X_t^{(k)}))^s}{\sum_{i=1}^J (\frac{\pi_1}{\pi_0}(X_t^{(i)}))^s}\bigg|_{s=0} \\
&= -\frac{(\frac{\pi_1}{\pi_0}(X_t^{(k)}))^s \log \frac{\pi_1}{\pi_0}(X_t^{(k)}) \sum_{i=1}^J (\frac{\pi_1}{\pi_0}(X_t^{(i)}))^s - (\frac{\pi_1}{\pi_0}(X_t^{(k)}))^s \sum_{i=1}^J (\frac{\pi_1}{\pi_0}(X_t^{(i)}))^s \log \frac{\pi_1}{\pi_0}(X_t^{(i)})}{\left(\sum_{i=1}^J (\frac{\pi_1}{\pi_0}(X_t^{(i)}))^s\right)^2}\bigg|_{s=0} \\
&= -\frac{J \log \frac{\pi_1}{\pi_0}(X_t^{(k)}) - \sum_{i=1}^j \log \frac{\pi_1}{\pi_0}(X_t^{(i)})}{J^2} = -\frac{1}{J}\left(\log \frac{\pi_1}{\pi_0}(X_t^{(k)}) - \frac{1}{J}\sum_{i=1}^J \log \frac{\pi_1}{\pi_0}(X_t^{(i)})\right).
\end{aligned}$$

Hence, the ODE arising from the limit of (17) as $\Delta t \to 0$ is the KFRFlow interacting particle system (9)

$$\dot{X}_t^{(j)} = \begin{pmatrix}\nabla_1 K(X_t^{(j)}, X_t^{(1)}) & \cdots & \nabla_1 K(X_t^{(j)}, X_t^{(J)})\end{pmatrix} M_t^{-1} \frac{1}{J} \sum_{k=1}^J \left(\log \frac{\pi_1}{\pi_0}(X_t^{(k)}) - \frac{1}{J}\sum_{i=1}^J \log \frac{\pi_1}{\pi_0}(X_t^{(i)})\right) \begin{pmatrix} K(X_t^{(k)}, X_t^{(1)}) \\ \vdots \\ K(X_t^{(k)}, X_t^{(J)})\end{pmatrix},$$

with initial condition $\{X_0^{(j)}\}_{j=1}^J \overset{\text{i.i.d.}}{\sim} \pi_0$.

B. Kernel Fisher–Rao Diffusion

The continuity equation (3) which we solve for the velocity v_t can equivalently be written for any $\epsilon > 0$ as a Fokker–Planck equation

$$\partial_t \pi_t = -\nabla \cdot (\pi_t (v_t + \epsilon \nabla \log \pi_t)) + \epsilon \nabla^2 \pi_t,$$

corresponding to the SDE

$$\mathrm{d}X_t = (v_t(X_t) + \epsilon \nabla \log \pi_t(X_t))\,\mathrm{d}t + \sqrt{2\epsilon}\,\mathrm{d}W_t, \quad t \in [0,1]. \tag{27}$$

This SDE possesses the same marginal distributions as the ODE (2). Using the same interacting particle approximation for the mean-field velocity v_t (8) that we use to define deterministic KFRFlow (9), we can also define a *stochastic* interacting particle system for approximately traversing the geometric mixture $\pi_t \propto \pi_0^t \pi_1^{1-t}$,

$$\mathrm{d}X_t^{(j)} = \nabla \mathbf{K}_t(X_t^{(j)})^\top M_t^{-1} \frac{1}{J} \sum_{k=1}^J \left(\log \frac{\pi_1}{\pi_0}(X_t^{(k)}) - \frac{1}{J}\sum_{i=1}^J \log \frac{\pi_1}{\pi_0}(X_t^{(i)})\right) \mathbf{K}_t(X_t^{(k)})\,\mathrm{d}t + \epsilon \nabla \log \pi_t(X_t^{(j)})\,\mathrm{d}t + \sqrt{2\epsilon}\,\mathrm{d}W_t, \tag{28}$$

where we have used the notation $\mathbf{K}_t(\cdot) = (K(\cdot, X_t^{(1)}), \dots, K(\cdot, X_t^{(J)}))^\top$ and $\nabla \mathbf{K}_t \in \mathbb{R}^{J \times d}$ is the Jacobian of $\mathbf{K}_t$. Equation (28) is obtained by applying Monte Carlo approximations to v_t in the SDE (27), where v_t is as in (8). Because we know $\pi_t \propto \pi_0^{1-t}\pi_1^t$ explicitly, the score of π_t can be computed directly as $\nabla \log \pi_t = (1-t)\nabla \log \pi_0 + t \nabla \log \pi_1$. We refer to the interacting particle system (28) as **Kernel Fisher–Rao Diffusion** (KFRD) and note that it can be simulated with any off-the-shelf SDE solver, for example Euler–Maruyama.

C. Additional Numerical Results

C.1. Two-Dimensional Bayesian Posteriors

C.1.1. Experimental setup

We apply KFRFlow (9), KFRFlow-I (17), and KFRD to sample three two-dimensional densities. In all three cases π_1 is a Bayesian posterior proportional to $\pi_0\ell$ for a likelihood $\ell = \pi(y^* \mid \cdot)$ of the form

$$\ell(x) \propto \exp\left(-\frac{1}{\sigma_\varepsilon^2}\|y^* - G(x)\|_2^2\right),$$

i.e., $y^* \in \mathbb{R}$ is Gaussian with mean $G(x)$ and variance σ_ε^2. Definitions of the three likelihoods and descriptions of the deformation behavior they entail may be found in Table 1.

Table 1. Likelihoods for the two-dimensional Bayesian example problems

$G(x)$	y^*	σ_ϵ^2	Behavior	Nickname
$\sqrt{x_1^2 + x_2^2}$	2	0.25^2	Concentration	Donut
$\sin(x_2)$ $+$ $\cos(x_1)$	-1	0.6^2	Bimodality	Butterfly
$\sin(x_1x_2)$ $+$ $\cos(x_1x_2)$	-1	0.5^2	Multimodality	Spaceships

We compare the sampling performances of KFRFlow, KFRFlow-I, and KFRD to those of EKI, EKS, CBS, SVGD, and ULA. We use the algorithms to generate target ensembles of size $J \in \{25, 50, 100, 200, 400\}$ with number of steps $N \in \{2^1, 2^2, \ldots, 2^8\}$. For unit-time KFRFlow(-I), KFRD, and EKI, the resulting step-size is $1/N$, but for infinite-time EKS, CBS, SVGD, and ULA we must choose a stopping time $T > 0$, resulting in a step-size of T/N. We test a range of stopping times T for EKS, CBS, SVGD, and ULA, regularization levels λ for KFRFlow(-I), noise levels ϵ for KFRD, and temperature β for CBS and report KSD corresponding to the best parameter settings for each (J, N). We do not inflate M_t in KFRD in these two-dimensional examples (i.e., $\lambda = 0$ for KFRD). The resulting choices of parameters for each algorithm and setting of (J, N) can be seen in Figure 8.

We use a fourth-order Adams–Bashforth discretization of KFRFlow because we find that it generates better samples than forward Euler at little additional cost, while by contrast we use forward Euler for SVGD because we find that SVGD does not benefit from multistep discretizations in these examples. For comparisons of the sampling performance of these different ODE discretization methods, see Figures 9 and 10. We use Euler–Maruyama discretizations for KFRD, EKI, EKS, CBS, and ULA. The values of KSD we report are averages over 30 trials.

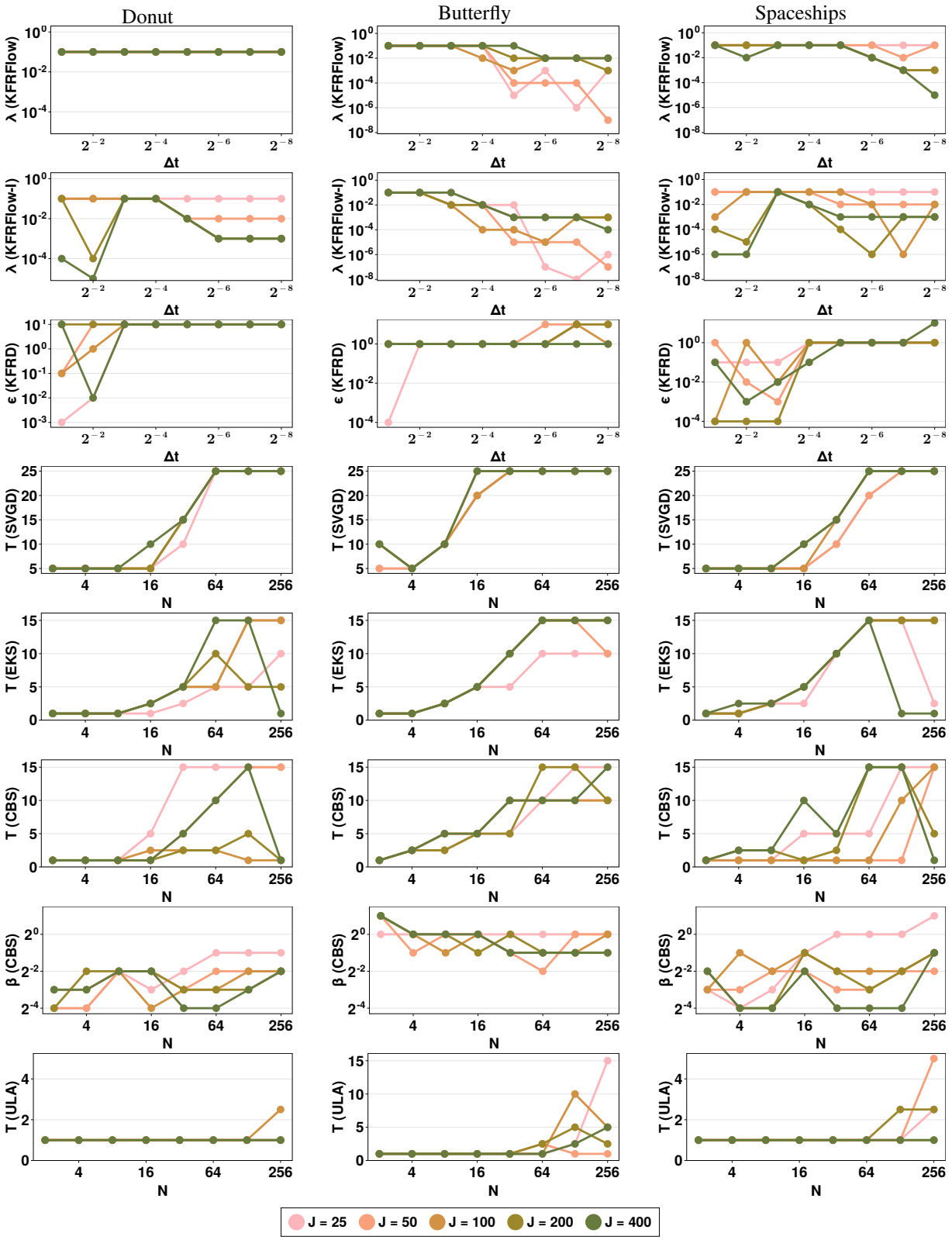

Figure 8. Optimal parameter choices for (from top to bottom) KFRFlow, KFRFlow-I, KFRD, SVGD, EKS, CBS, and ULA for the donut (left), butterfly (middle), and spaceships (right) examples.

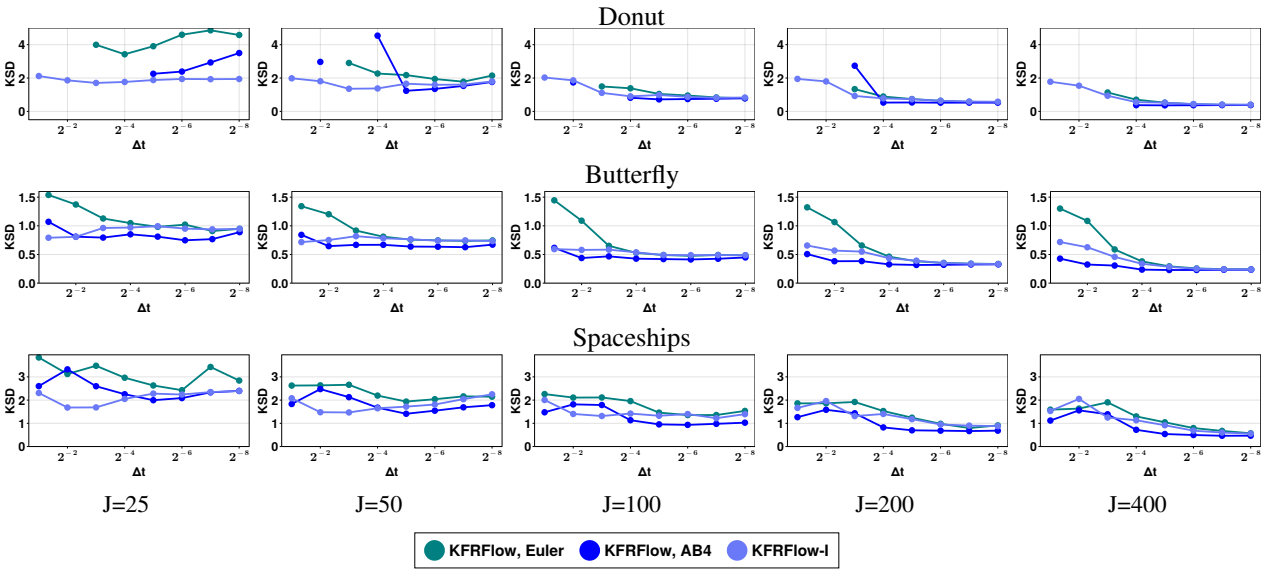

Figure 9. **Two-dimensional posteriors:** average KSD of ensembles generated by forward Euler and AB4 discretizations of KFRFlow and by KFRFlow-I for varying ensemble size and Δt. KFRFlow-I and the AB4 discretization of KFRFlow generally outperform the forward Euler discretization of KFRFlow.

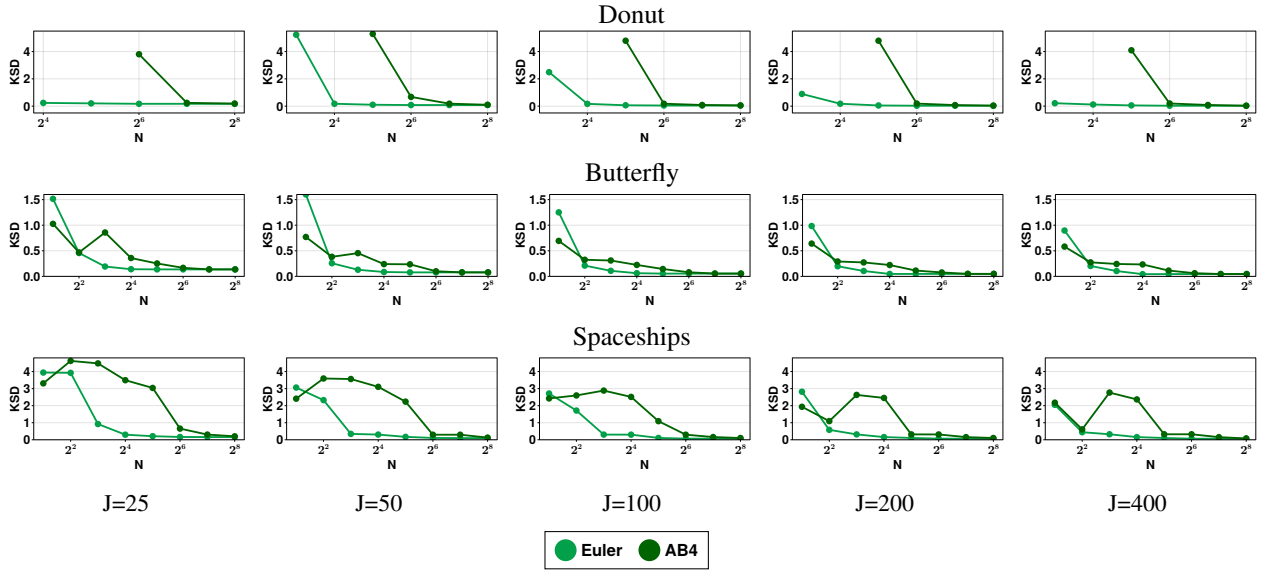

Figure 10. **Two-dimensional posteriors:** average KSD of ensembles generated by forward Euler and AB4 discretizations of SVGD. Using AB4 in place of forward Euler with SVGD tends only to make things worse.

C.1.2. Extended Values of J

In Figures 11 and 12 we compare the quality of samples generated by KFRFlow and KFRFlow-I to those of EKI, EKS, and CBS, and the quality of samples generated by KFRD to those of SVGD, and ULA. In Figure 13 we show the evolution of KSD between π_1 and samples generated by the unit-time methods KFRFlow, KFRFlow-I, KFRD, and EKI as a function of t, with $\Delta t = 2^{-8}$. These figures contain results for a wider range of J than the main-body Figures 3 to 5. In these and following figures, results corresponding to gradient-free methods are plotted with solid lines and circles, while results corresponding to gradient-based methods are plotted with dashed lines and diamonds.

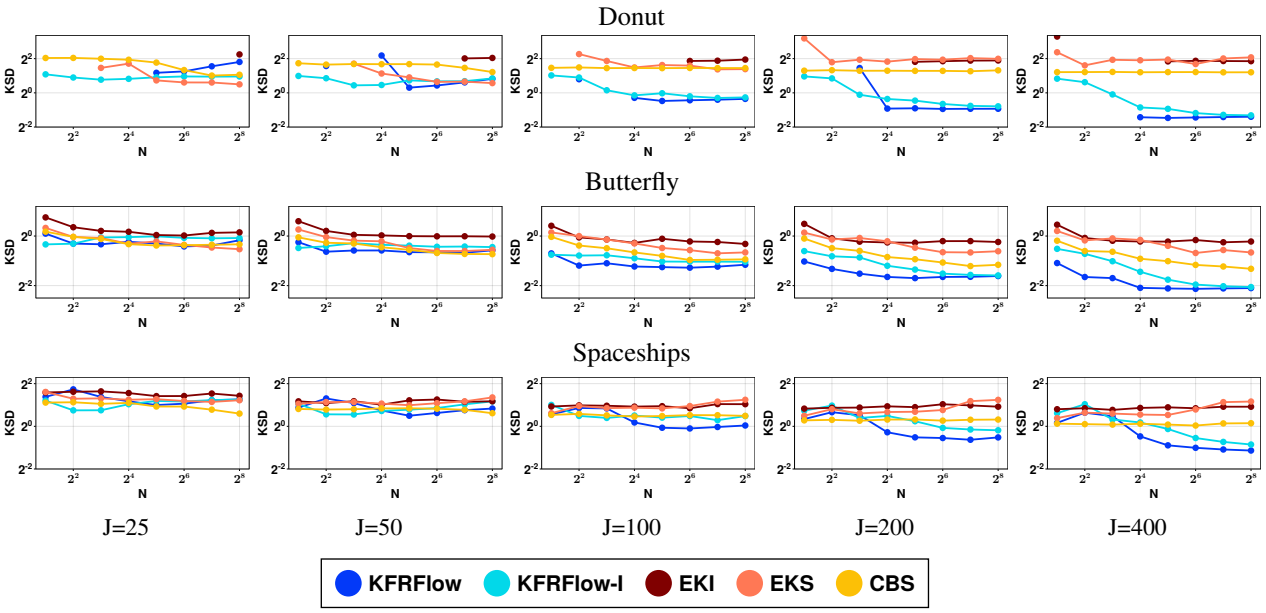

Figure 11. **Two-dimensional posteriors:** average KSD at stopping time between π_1 and ensembles of size $J \in \{25, 50, 100, 200, 400\}$ generated by gradient-free samplers. A missing point indicates that a method was unstable at that setting of N.

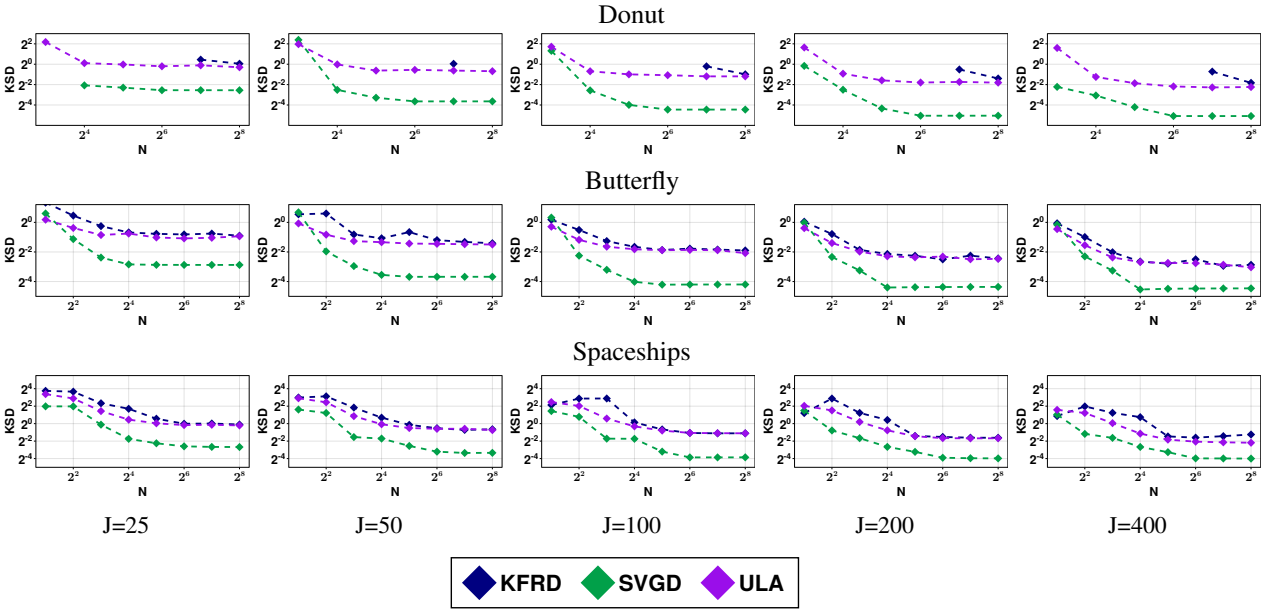

Figure 12. **Two-dimensional posteriors:** average KSD at stopping time between π_1 and ensembles of size $J \in \{25, 50, 100, 200, 400\}$ generated by gradient-based samplers. A missing point indicates that a method was unstable at that setting of N.

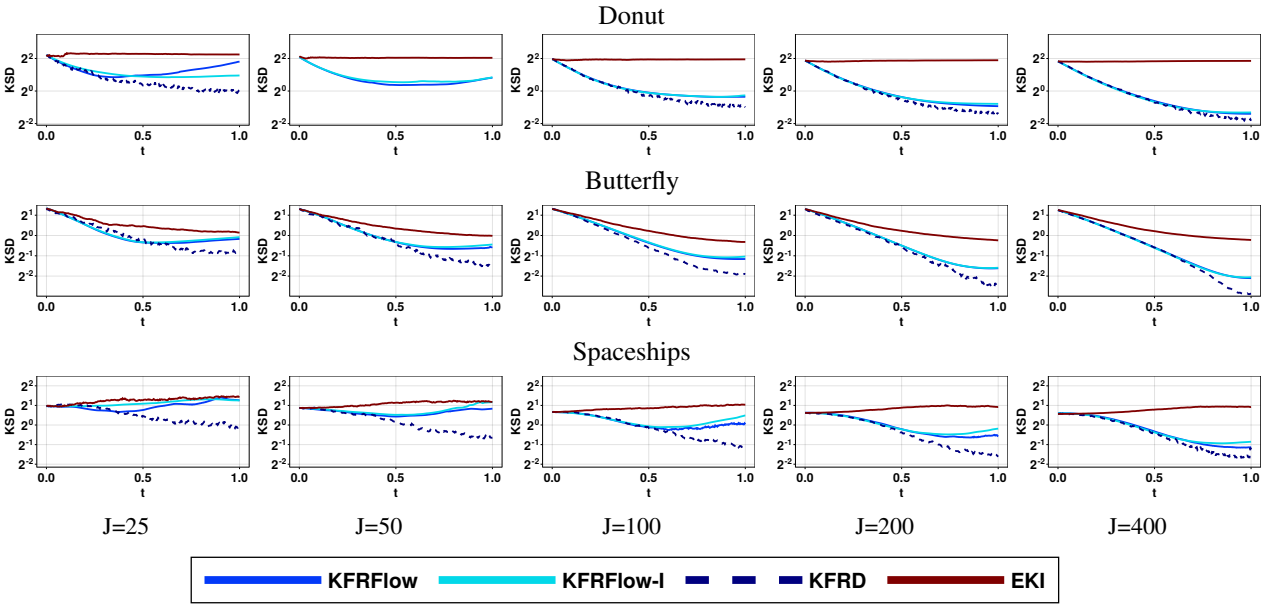

Figure 13. **Two-dimensional posteriors:** evolution of KSD between π_1 and samples generated by the unit-time methods KFRFlow, KFRFlow-I, KFRD, and EKI with $t \in [0, 1]$ for ensembles of size $J \in \{25, 50, 100, 200, 400\}$ and $\Delta t = 2^{-8}$.

C.1.3. KSD BETWEEN SAMPLES AND INTERMEDIATE DISTRIBUTIONS

In Figure 14 we plot the KSD between samples $\{X_t^{(j)}\}_{j=1}^J$ generated by KFRFlow, KFRFlow-I, and KFRD and the intermediate distributions $\pi_t \propto \pi_0^{1-t}\pi_1^t$, for $t \in [0, 1]$. $\mathrm{KSD}(\pi_t, \{X_t^{(j)}\}_{j=1}^J)$ can be viewed as a sort of "discretization error": even at $t = 0$ we see that the KSD between π_0 and the samples $\{X_0^{(j)}\}_{j=1}^J$, which are taken directly from $\pi_0 = \mathcal{N}(0, I_d)$, is nonzero due to finite J.

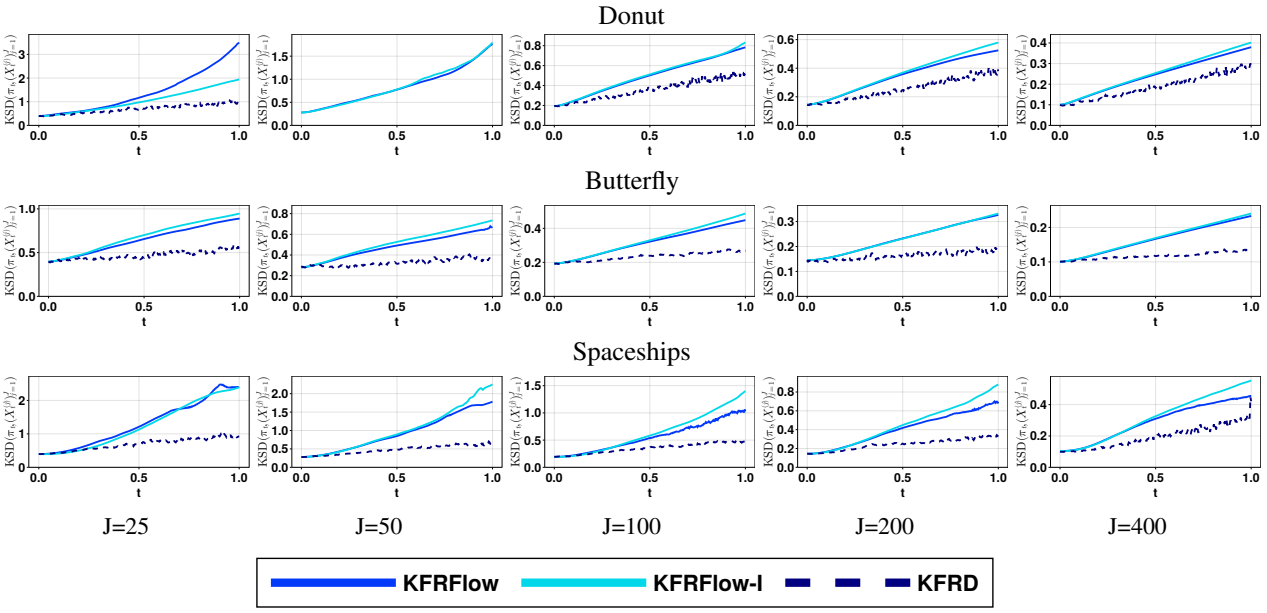

Figure 14. **Two-dimensional posteriors:** KSD between intermediate distributions π_t and samples at time t generated by KFRFlow, KFRFlow-I, and KFRD for ensemble sizes $J \in \{25, 50, 100, 200, 400\}$ and $\Delta t = 2^{-8}$. A missing line in a plot indicates that the method was unstable at that setting of J.

One naturally expects $\mathrm{KSD}(\pi_t, \{X_t^{(j)}\}_{j=1}^J)$ to increase in time owing to error incurred from kernelization of the solution to (4), Monte Carlo approximation of the mean-field ODE (8), and time-discretization of the finite-sample ODE (9), and we indeed see such increases in Figure 14. The increases are lower for KFRD than KFRFlow and KFRFlow-I, perhaps due to the presence of gradient information in KFRD. Here the increases generally appear to be approximately linear in time.

C.1.4. Comparison to random walk Metropolis

In Figure 15 we compare the quality of samples generated by KFRFlow and KFRFlow-I to that of samples generated by random walk Metropolis (RWM, Robert & Casella (2004)). We use RWM to sample the target distributions in both "serial" mode, in which a single long chain is generated and the last J states of this chain are retained as samples from the target, and "parallel" mode, in which J independent chains are run and the last state from each chain is taken to form a set of J samples from π_1. When we compare the sampling performances of serial and parallel RWM to KFRFlow and KFRFlow-I we compare for equivalent *total* numbers of steps; that is, we run N steps of KFRFlow and KFRFlow-I, N steps on each of the J chains in parallel RWM, and and NJ steps on the single chain in serial RWM. For both RWM settings we tune the variance of the isotropic Gaussian proposal distribution to attain the optimal acceptance rate of 23% (Yang et al., 2020).

Though serial mode is the generally the setting of choice for RWM, owing to the fact that parallel mode inefficiently replicates transient ("burn-in") behavior across chains, we see in Figure 15 that for sufficiently large N, parallel RWM produces better-quality samples than serial RWM, KFRFlow, and KFRFlow-I. For these large values of N, parallel RWM benefits from multiple starting points, which help it collectively sample all regions of the target distributions, and from the fact that each individual chain has moved adequately by step N; these particular target distributions do not require long burn-in times. By contrast, for small N we deduce from Figure 15 that the J parallel chains are often not adequately burnt in, as serial RWM (with its J-fold longer burn-in time) and KFRFlow-I generally produce better samples in this setting. For most instances of $N > 8$ steps, KFRFlow or KFRFlow-I produces samples of comparable or better quality than serial RWM, as these interacting particle algorithms are able to explore the target distributions more effectively than a single chain of RWM. For $N \leq 8$ the resulting Δt is often too large for KFRFlow and KFRFlow-I to be integrated accurately, and serial RWM benefits from the fact that it has $(N-1)J$ burn-in steps.

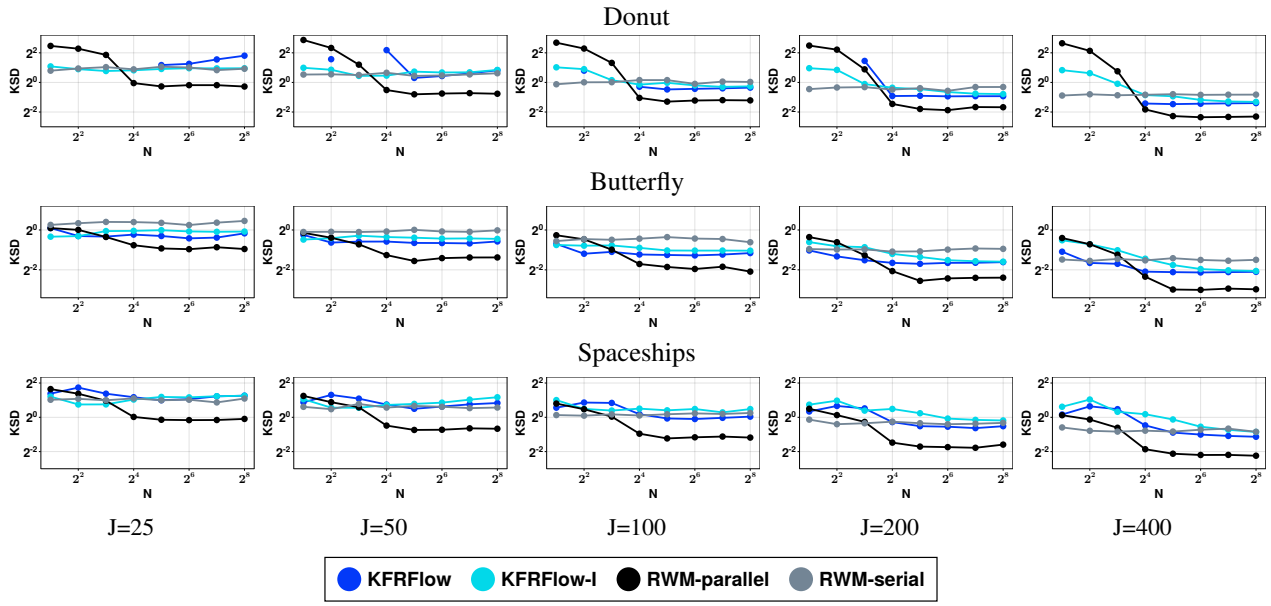

Figure 15. **Two-dimensional posteriors:** average KSD at stopping time between π_1 and ensembles of size $J \in \{25, 50, 100, 200, 400\}$ generated by KFRFlow, KFRFlow-I, and RWM. A missing point indicates that a method was unstable at that setting of N.

C.1.5. Effect of Δt

In Figure 16 we investigate the impact of step-size Δt on the evolution of sample quality, as measured by KSD, with $t \in [0, 1]$. For each example and setting of $\Delta t \in \{2^{-1}, 2^{-2}, \ldots, 2^{-8}\}$ we generate $J = 300$ approximate samples of π_1

with KFRFlow and KFRFlow-I and compute KSD between the samples and π_1 at each step of the iterations. We regularize M_t in the Euler discretization of KFRFlow with λ set to 10^{-1}, 10^{-8}, and 10^{-11} for the donut, butterfly, and spaceships examples, respectively, but do not regularize M_t in KFRFlow-I. The data plotted in Figure 16 are the result of averaging the values of KSD over 30 repeated trials at each setting of Δt.

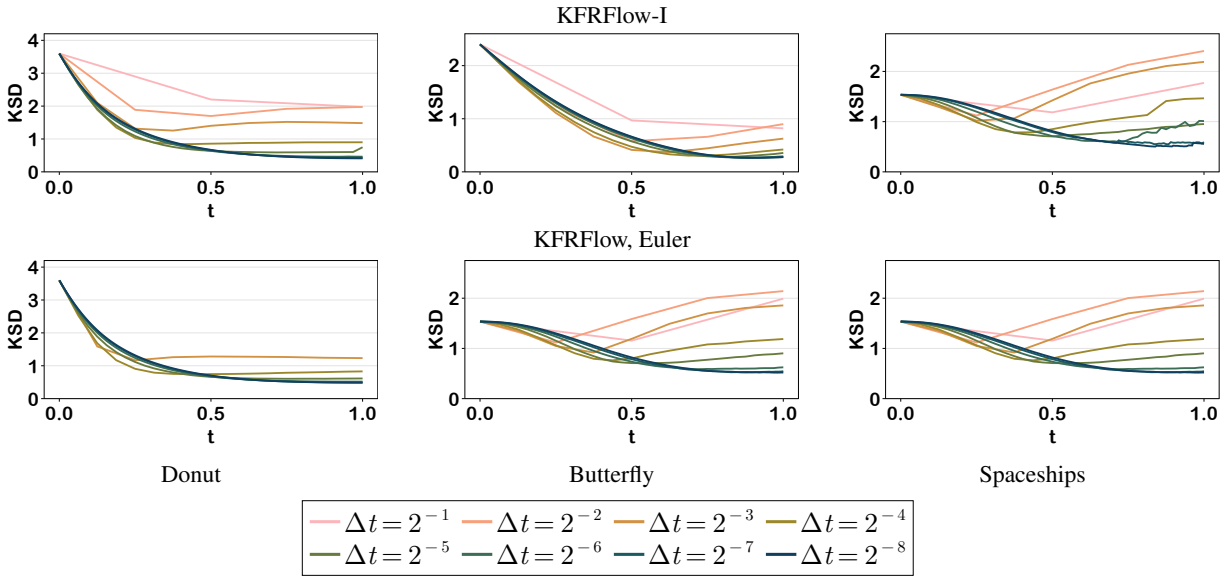

Figure 16. **Two-dimensional posteriors:** KSD between π_1 and samples generated by KFRFlow-I (top) and an explicit Euler discretization of KFRFlow (bottom) versus t for various Δt. In each example Δt must be below a certain threshold to ensure that KSD decreases monotonically throughout the iteration. KFRFlow-I is apparently more stable at large Δt than the Euler discretization of KFRFlow.

C.1.6. Empirical Runtime

In Figure 17 we display the median time taken to compute one update (i.e., transform $\{X_t^{(j)}\}_{j=1}^J$ into $\{X_{t+\Delta t}^{(j)}\}_{j=1}^J$) in each of KFRFlow, KFRD, EKI, EKS, CBS, SVGD, and ULA as a function of ensemble size J in the two-dimensional setting. Benchmarks were performed in Julia using `BenchmarkTools.jl` (Chen & Revels, 2016) on a 2020 MacBook Air with Apple M1 processor.

We see in Figure 20 that the runtimes increase polynomially with ensemble size J, as one would expect based on, e.g., the complexity of KFRFlow (Section 5.2), and that the methods can be organized into three clusters based on cost: KFRFlow, KFRD, and SVGD are most expensive, EKI and EKS are cheapest, and CBS and ULA fall somewhere in between. As $\nabla \log \pi_1$ is cheap to evaluate in these examples, the data in Figure 17 do not capture extra costs that may be incurred by gradient-based methods (SVGD, KFRD, and ULA) in the setting where $\nabla \log \pi_1$ is expensive.

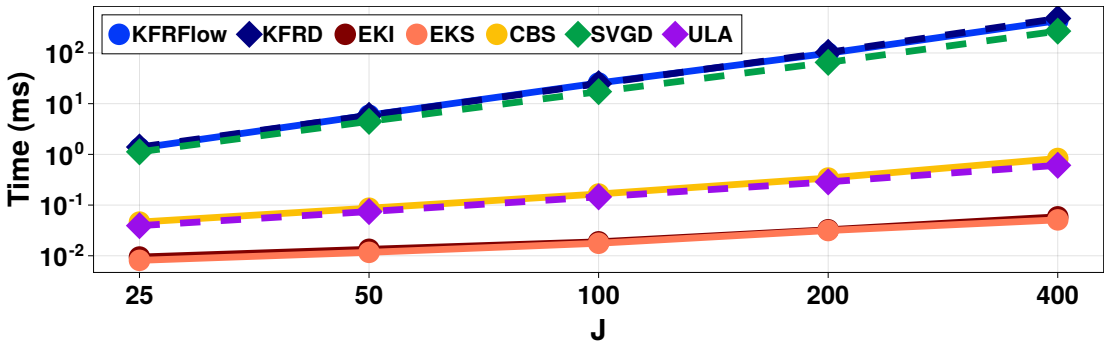

Figure 17. **Two-dimensional posteriors:** time (ms) taken to compute one ensemble update for each of KFRFlow, KFRD, EKI, EKS, CBS, SVGD, and ULA in two dimensions as a function of ensemble size J.

C.2. Higher-Dimensional Funnels

C.2.1. Experimental Setup

For dimension $d \in \{5, 10, 15, 20\}$ we compare the performance of KFRFlow-I, KFRD, CBS, SVGD, and ULA in sampling from "funnel" distributions of the form

$$\pi_1(\mathbf{x}) = \mathcal{N}(x_1; 0, 9)\mathcal{N}(\mathbf{x}_{2:d}; \mathbf{0}, \exp(x_1)\mathbf{I}),$$

i.e., X_1 is distributed normally with mean zero and variance nine, and $(X_2, \ldots, X_d)$ are multivariate normal with mean zero and covariance matrix $\exp(X_1)\mathbf{I}$. This family of distributions appears in Neal (2003) and is commonly used as a benchmark for sampling algorithms, e.g., Arbel et al. (2021); Zhang et al. (2023); Xu et al. (2023b).

For each setting of d we apply these algorithms to generate $J = 100$ samples from π_1. For KFRFlow-I and KFRD we set $\Delta t = 0.01$, corresponding to $N = 100$ steps for the infinite-time algorithms CBS, SVGD, and ULA. As in Section 6.1 we optimize the hyperparameters for KFRFlow-I, KFRD, CBS, SVGD, and ULA via coarse direct search to minimize KSD between the final samples and π_1. The resulting hyperparameter values are shown in Table 2. The data in Figures 6 to 7, 18 and 19 are produced by averaging the results of 30 independent trials.

	$d = 5$	$d = 10$	$d = 15$	$d = 20$
λ (KFRFlow-I)	0.01	0.001	0.001	0.001
ϵ (KFRD)	5	5	5	2.5
λ (KFRD)	0.001	0.01	0.1	0.1
T (CBS)	25	12.5	25	25
β (CBS)	0.125	0.5	0.25	0.25
T (SVGD)	100	100	100	100
T (ULA)	12.5	12.5	25	12.5

Table 2. **Funnels:** Selected hyperparameters for each algorithm and target dimension d.

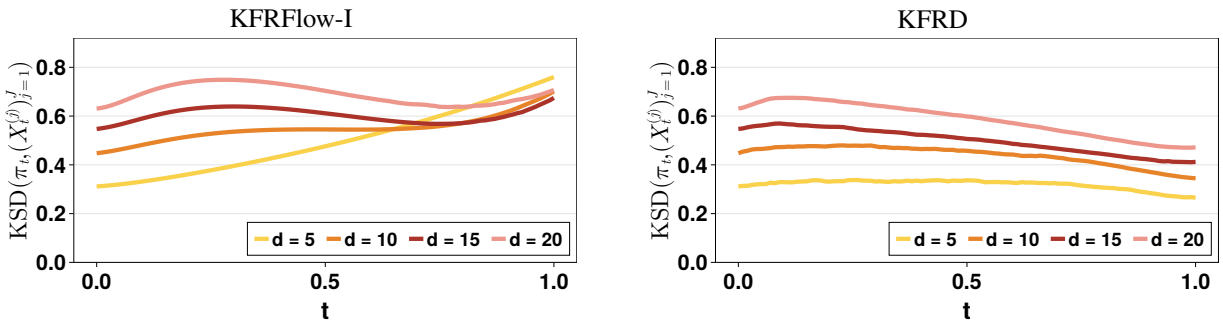

Figure 18. **Funnels:** KSD between samples at time t and the intermediate distribution π_t, for $t \in [0, 1]$; KFRFlow-I (left) and KFRD (right). This "discretization error" somewhat surprisingly seems to decrease with t for KFRD but is noticeably non-monotonic for KFRFlow-I.

C.2.2. KSD between Samples and Intermediate Distributions

In Figure 18 we plot the KSD between samples $\{X_t^{(j)}\}_{j=1}^J$ generated by KFRFlow and KFRD and the intermediate distributions $\pi_t \propto \pi_0^{1-t}\pi_1^t$, $t \in [0, 1]$. This quantity can be viewed as a sort of "discretization error," for even at $t = 0$ we see that the KSD between π_0 and the samples $\{X_0^{(j)}\}_{j=1}^J$, which are sampled directly from $\pi_0 = \mathcal{N}(0, I_d)$, is nonzero due to the finiteness of J.

As mentioned in Appendix C.1.3, one naturally expects $\mathrm{KSD}(\pi_t, \{X_t^{(j)}\}_{j=1}^J)$ to increase in time due to accumulation of error, but interestingly we see in Figure 18 that $\mathrm{KSD}(\pi_t, \{X_t^{(j)}\}_{j=1}^J)$ is not generally monotone increasing in time.

This quantity, in fact, is mostly decreasing in time for KFRD and tends to follow an undulating pattern, which becomes more evident with increasing d, for KFRFlow-I. Understanding these phenomena and their relationship to choice of IPS (KFRFlow-I vs KFRD) and dimension d is an interesting area for future work.

C.2.3. Comparison to Random Walk Metropolis

In Figure 19 we compare the quality of samples generated by KFRFlow-I to that of samples generated by random walk Metropolis (RWM, Robert & Casella (2004)). As in Appendix C.1.4, we apply RWM in both serial mode and parallel mode and make comparisons among algorithms for equivalent total numbers of steps. For both RWM settings we tune the variance of the isotropic Gaussian proposal distribution to attain the optimal acceptance rate of 23% (Yang et al., 2020).

Similarly to the behavior in Figure 15, we see in Figure 19 that for all settings of d parallel RWM produces better samples, as measured with KSD, than serial RWM. We posit that the burn-in time for RWM is not very long for these funnel distributions, and thus parallel RWM is benefiting from multiple initializations and from the fact that each individual chain is adequately burnt in after $N = 100$ steps. Interestingly, we see in Figure 19 that the quality of samples produced by RWM in both modes degrades with dimension but that the same is not true of KFRFlow-I: for $d = 5$ KFRFlow-I produces samples that are slightly worse than serial RWM, but for $d > 5$ the samples from KFRFlow-I are better than those from serial RWM, with the quality of samples produced by KFRFlow-I becoming comparable to that of samples from parallel RWM by $d = 20$.

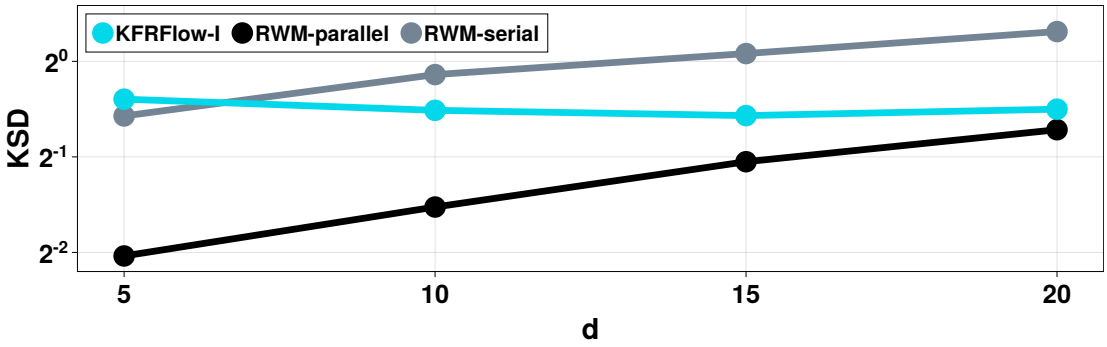

Figure 19. **Funnels:** average KSD at stopping time between π_1 and samples generated by KFRFlow-I and RWM for $d \in \{5, 10, 15, 20\}$

C.2.4. Empirical Runtime

In Figure 20 we display the median time taken to compute one update (i.e., transform $\{X_t^{(j)}\}_{j=1}^J$ into $\{X_{t+\Delta t}^{(j)}\}_{j=1}^J$) in each of KFRFlow-I, KFRD, CBS, SVGD, and ULA for the settings in the funnel example. Benchmarks were performed in Julia using `BenchmarkTools.jl` (Chen & Revels, 2016) on a 2020 MacBook Air with Apple M1 processor.

We see in Figure 20 that the runtimes do not demonstrate distinct dependence on dimension d and that the runtimes of KFRFlow-I, KFRD, and SVGD are comparable, with those of CBS and ULA being significantly lower. Given that the sampling performance of ULA, SVGD, and KFRD were essentially the same in this example, cost considerations suggest that ULA is the best choice of gradient-based method here, but for gradient-free samplers the situation is more nuanced: CBS is cheaper than KFRFlow-I, but KFRFlow-I produces better samples.

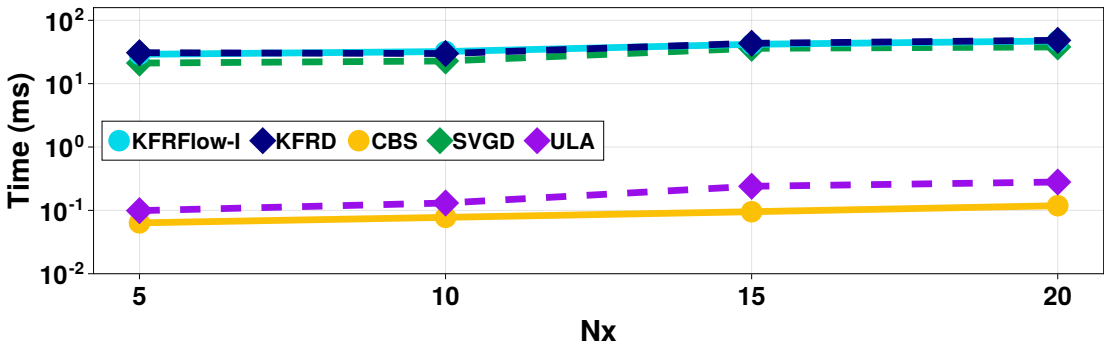

Figure 20. **Funnels:** median time (ms) taken to compute one ensemble update in each of KFRFlow-I, KFRD, CBS, SVGD, and ULA for the in the funnel example as a function of dimension d. Gradient-free methods are plotted with solid lines and circles, while gradient-based methods are plotted with dashed lines and diamonds.