Accelerated First-order Methods on the Wasserstein Space for Bayesian Inference

Chang Liu[1]*, **Jingwei Zhuo**[1], **Pengyu Cheng**[2], **Ruiyi Zhang**[2], **Jun Zhu**[1]†, **Lawrence Carin**[2]†
[1] Dept. of Comp. Sci. & Tech., TNList Lab, State Key Lab for Intell. Tech. & Sys.,
[1] Tsinghua University, Beijing, 100084, China
[2] Duke University

Abstract

We consider doing Bayesian inference by minimizing the KL divergence on the 2-Wasserstein space $\mathcal{P}_2$. By exploring the Riemannian structure of $\mathcal{P}_2$, we develop two inference methods by simulating the gradient flow on $\mathcal{P}_2$ via updating particles, and an acceleration method that speeds up all such particle-simulation-based inference methods. Moreover we analyze the approximation flexibility of such methods, and conceive a novel bandwidth selection method for the kernel that they use. We note that $\mathcal{P}_2$ is quite abstract and general so that our methods can make closer approximation, while it still has a rich structure that enables practical implementation. Experiments show the effectiveness of the two proposed methods and the improvement of convergence by the acceleration method.

1 Introduction

Bayesian inference is an important topic in the machine learning community for its power in modeling and reasoning uncertainty among data. Its task is to get access to the posterior distribution p given data for a Bayesian model. As p is intractable for general Bayesian models, various methods are developed for approximation. Variational inference methods (VIs) try to approximate p by a tractable distribution q that minimizes the Kullback-Leibler divergence (KLD) to p. A parametric distribution family is typically chosen for tractability (e.g. [33, 11]) and the problem can be efficiently solved by classical optimization methods. But this restricts the flexibility of the approximation thus the closeness to p suffers. Markov chain Monte Carlo methods (MCMCs) (e.g. [9, 23, 35, 4]) aim to directly draw samples from p. Although they are asymptotically accurate, a large sample size is required due to the autocorrelation between samples, which makes a big cost at test time.

Recently, there appear methods that minimize the KLD on distribution spaces that are more abstract and general than parametric families. In algorithmic appearance, such methods iteratively update a set of samples, or particles, so that the set of particles gradually becomes representative for p. They can achieve a better approximation than classical VIs because of the greater flexibility of particles over parametric forms, and they are more particle-efficient than MCMCs, in the sense that a better approximation can be achieved with the same sample size, since they take the particle interaction into account and focus on finite-particle performance. Stein Variational Gradient Descent (SVGD) [18] is a remarkable instance. It updates particles along the gradient flow (GF) (steepest descending curves) of the KLD on the distribution manifold $\mathcal{P}_{\mathcal{H}}$, whose tangent space is the reproducing kernel Hilbert space of a kernel [17]. Its unique benefits make it a popular method: it has been modified for Riemannian support space [16] and structured posterior [39], and applied to deep generative models [34, 27] and Bayesian reinforcement learning [19, 10]. Another instance is the particle optimization method (PO) [3] where particles are updated by solving an optimization problem known as the variational formulation of the Langevin dynamics [12]. The algorithm resembles the Polyak's momentum [26]

*This work was done when the author was visiting Duke University.
†JZ and LC are corresponding authors.

Preprint. Work in progress.

version of SVGD. We call such methods particle-simulation-based variational inference methods (PSVIs) to highlight the difference to VIs and MCMCs.

In this work, we propose to minimize the KLD on the 2-Wasserstein space $\mathcal{P}_2$, which is general enough to contain all distributions with finite second-order moment, while still tractable for computation thanks to its Riemannian structure. We contribute to the field of PSVI in three folds: (i) We make a theoretical analysis on the flexibility of PSVIs and find out that the particle update rule can be expressed by either a density function or an optimization problem over a family of test functions, and PSVIs have the flexibility up to either a smoothed density or a smoothed test function family. Typically kernel is used to satisfy this restriction (Section 3.1). (ii) We propose two PSVIs: Gradient Flow with Smoothed Density / test Function (GFSD/GFSF), by simulating the GF of the KLD on $\mathcal{P}_2$ (Section 3.2). We also develop a principled bandwidth selection method (Section 3.2.3) for them based on their relation to the Langevin dynamics (Remark 4). (iii) We develop an acceleration method for PSVIs based on the Riemannian version [20] of the Nesterov's acceleration method [24], by studying the exponential map and the parallel transport on $\mathcal{P}_2$ (Section 4). The effectiveness of GFSD/GFSF and the acceleration effect is observed in experiments.

Related work As mensioned, SVGD simulates the GF on $\mathcal{P}_{\mathcal{H}}$, while ours on $\mathcal{P}_2$. We note in Remark 3 that $\mathcal{P}_2$ is a more natural and well-defined manifold of distributions, while $\mathcal{P}_{\mathcal{H}}$ is not guaranteed for the existence of the tangent vector of every smooth curve on it. Moreover, $\mathcal{P}_2$ provides essential ingredients to develop the bandwidth selection method as well as the acceleration method. Algorithmically, GFSD/GFSF does not average gradients for each particle, enabling every particle to find the high-probability region near it more efficiently in early stage.

The idea of the PO method is essentially to simulate the GF of the KLD on $\mathcal{P}_2$ as well, but their simulation is based on the discretization using the minimal movement scheme (MMS) ([1], Def 2.0.6) on $\mathcal{P}_2$, while ours is based on simulating the associated dynamics on the support space (Section 2.1), which has an intuitive physics picture and easy to implement. To find the desired tangent vector on $\mathcal{P}_2$, the PO method adds a noise to the SVGD tangent vector in $\mathcal{P}_{\mathcal{H}}$, which is unnatural and may affect convergence. Although the PO algorithm takes the form of Polyak's momentum acceleration [26] of SVGD for each particle, it requires further theoretical interpretation to be an actual acceleration method, since SVGD is not an optimization method for each particle. Additionally, Nesterov's acceleration is known to be more stable than Polyak's momentum [31].

2 Preliminaries

In this part, we briefly introduce the Wasserstein space as a Riemannian manifold and the gradient flow on it, as well as the theory of SVGD. We use bold symbols (e.g. $\boldsymbol{q}$) to represent probability measures and regular ones (e.g. q) for the corresponding density functions when exist.

2.1 $\mathcal{P}_2$ as a Riemannian Manifold

The concept of Wasserstein space rises from the optimal transport problem. Let $\mathcal{X} = \mathbb{R}^D$ be the support space, $d(\cdot,\cdot)$ be the typical Euclidean distance, and $\mathcal{P}(\mathcal{X})$ be the space of probability measures on $\mathcal{X}$. We define $\mathcal{P}_2(\mathcal{X}) := \{\boldsymbol{q} \in \mathcal{P}(\mathcal{X}) : \exists x_0 \in \mathcal{X} \text{ s.t. } \mathbb{E}_{\boldsymbol{q}}[d(x_0,x)^2] < +\infty\}; W_2(\boldsymbol{q},\boldsymbol{p}) := \left(\inf_{\boldsymbol{\pi}\in\Pi(\boldsymbol{q},\boldsymbol{p})} \mathbb{E}_{\boldsymbol{\pi}(x,y)}[d(x,y)^2]\right)^{1/2}, \forall \boldsymbol{q},\boldsymbol{p} \in \mathcal{P}(\mathcal{X})$, where $\Pi(\boldsymbol{q},\boldsymbol{p}) := \{\boldsymbol{\pi} \in \mathcal{P}(\mathcal{X}\times\mathcal{X}) : \boldsymbol{\pi}(\cdot,\mathcal{X}) = \boldsymbol{q}, \boldsymbol{\pi}(\mathcal{X},\cdot) = \boldsymbol{p}\}$ is the space of joint probability measures with $\boldsymbol{q}$ and $\boldsymbol{p}$ as the marginals. It is well known that $(\mathcal{P}_2(\mathcal{X}), W_2)$ is a metric space ([32], Def 6.4), which is called the 2-Wasserstein space, and W_2 is called the 2-Wasserstein distance. We use $\mathcal{P}_2$ to denote $(\mathcal{P}_2(\mathcal{X}), W_2)$ for brevity. More precisely, $\mathcal{P}_2$ is a length space ([32], Thm 7.21), which inspires the exploration for any possible Riemannian structures on $\mathcal{P}_2$ (e.g. [25]). It is an appealing task since a Riemannian structure provides rich ingredients that enable explicit calculation thus practical implementation of quantities of interests, e.g. the gradient of a function on $\mathcal{P}_2$.

In the sense of Riemannian manifold, a smooth curve defines a tangent vector at some point on it as the differentiation along the curve (e.g. [5], Def 2.6). Specifically, for an absolutely continuous (AC) curve (roughly a "smooth" curve) $(\boldsymbol{q}_t)_{t\in[0,1]}$, it defines a tangent vector $\dot{\boldsymbol{q}}_{t_0}$ at $\boldsymbol{q}_{t_0}$ as: $\dot{\boldsymbol{q}}_{t_0}[F] := \frac{\mathrm{d}}{\mathrm{d}h} F(\boldsymbol{q}_{t_0+h})\big|_{h=0}$ for any function F of the form $F(\boldsymbol{q}) = \int_{\mathcal{X}} f \mathrm{d}\boldsymbol{q}$ with $f \in \mathcal{C}_c^\infty(\mathcal{X})$ (compactly supported smooth function). All such $\dot{\boldsymbol{q}}_{t_0}$ form a vector space: the tangent space $T_{\boldsymbol{p}}\mathcal{P}_2$. On the other hand, another notion of differentiating $\boldsymbol{q}_t$ is the *weak derivative* $\partial_t \boldsymbol{q}_t$, which is the generalization of the typical derivative through the rule of integration by parts: $(\partial_t \boldsymbol{q}_t)[\varphi] := \int_0^1 \int_{\mathcal{X}} \varphi_t(x) \mathrm{d}(\partial_t \boldsymbol{q}_t) =$

$-\int_0^1 \int_{\mathcal{X}} (\partial_t \varphi_t(x)) \mathrm{d}\boldsymbol{q}_t \mathrm{d}t, \forall \varphi_t(x) \in \mathcal{C}_c^\infty([0,1]\times\mathcal{X})$. The two notions are related by considering $\dot{\boldsymbol{q}}_t[F]$ as a weak derivative: $(\dot{\boldsymbol{q}}_t[F])[g] = (\partial_t \boldsymbol{q}_t)[fg], \forall g \in \mathcal{C}_c^\infty([0,1])$. The weak derivative description brings the vector field representation of tangent vectors in $\mathcal{P}_2$:

Lemma 1 (Continuity Equation ([32], Thm 13.8; [1], Thm 8.3.1, Prop 8.4.5))**.** *Denote the Lebesgue measure on $\mathbb{R}$ as $\mathscr{L}^1$. For any AC curve $(\boldsymbol{q}_t)_{t\in[0,1]}$ on $\mathcal{P}_2$, there exists a time-dependent Borel vector field $v_t(x)$ such that for $\mathscr{L}^1$-a.e. $t \in [0,1]$, $v_t \in \overline{\{\nabla\varphi : \varphi \in C_c^\infty(\mathcal{X})\}}^{L^2(\boldsymbol{q}_t;\mathbb{R}^D)}$ (where the overline means closure), and the continuity equation $\partial_t \boldsymbol{q}_t + \nabla\cdot(v_t \boldsymbol{q}_t) = 0$ holds in the weak sense: $\int_0^1 \int_{\mathcal{X}} \big(\partial_t\varphi_t(x) + v_t(x)\cdot\nabla\varphi_t(x)\big)\mathrm{d}\boldsymbol{q}_t(x)\mathrm{d}t, \ \ \forall \varphi \in \mathcal{C}_c^\infty([0,1]\times\mathcal{X})$. Moreover, such a v is $(\boldsymbol{q}\times\mathscr{L}^1)$-a.e. unique.*

We denote $\dot{\boldsymbol{q}}_t \sim v_t$ (similarly for $\partial_t \boldsymbol{q}_t$) if both notions refer to a same tangent vector on $\mathcal{P}_2$, from their own perspectives. Furthermore, the above space of v_t is recognized as the tangent space $T_{\boldsymbol{q}}\mathcal{P}_2$ at $\boldsymbol{q}$ ([1], Def 8.4.1). With the inner product on $T_{\boldsymbol{q}}\mathcal{P}_2$: $\langle \xi, \zeta\rangle_{T_{\boldsymbol{q}}\mathcal{P}_2} := \langle \xi, \zeta\rangle_{L^2(\boldsymbol{q};\mathbb{R}^D)} = \int_{\mathcal{X}} \xi(x)\cdot\zeta(x)\mathrm{d}\boldsymbol{q}(x)$, the Riemannian distance recovers the metric-space-sense distance due to the Benamou-Brenier formula [2]: $W_2(\boldsymbol{q}_0,\boldsymbol{q}_1)^2 = \inf_{\boldsymbol{q}_t}\int_0^1 \|\dot{\boldsymbol{q}}_t\|_{T_{\boldsymbol{q}_t}\mathcal{P}_2}\mathrm{d}t$. Now $\mathcal{P}_2$ can be treated as a Riemannian manifold.

The vector field representation has a solid physical intuition. Consider at time $t = 0$ there is a set of particles $\{x_0^{(i)}\}_i$ that distributes as $\boldsymbol{q}_0$. Let the i-th particle move with velocity $v_t(x_t^{(i)})$ for any moment $t \geq 0$. If $\dot{\boldsymbol{q}}_t \sim v_t$, $\{x_t^{(i)}\}_i$ will distribute as $\boldsymbol{q}_t$. This intuition can be formally stated:

Lemma 2 (Simulating a curve on $\mathcal{P}_2$ ([1], Prop 8.1.8))**.** *Let $(\boldsymbol{q}_t)_{t\in[0,1]}$ be an AC curve of AC measure (wrt the Lebesgue measure on $\mathbb{R}^D$, if not specified). Then for $\boldsymbol{q}_0$-a.e. $x \in \mathcal{X}$, there exists a globally defined differentiable map $X : [0,1]\times\mathcal{X} \to \mathcal{X}$ such that $\partial_t X_t(x) = v_t(x)$ where $v_t \sim \dot{\boldsymbol{q}}_t$, and $\boldsymbol{q}_t = (X_t)_{\#}\boldsymbol{q}_0$, which means for any Borel subset $\Omega \subset \mathcal{X}$, $\boldsymbol{q}_t(\Omega) = \boldsymbol{q}_0(X_t^{-1}(\Omega))$.*

2.2 Gradient Flows on $\mathcal{P}_2$

There are various ways to define the gradient flow (GF) of a function on metric spaces (e.g. [1], Def 11.1.1; [32], Def 23.7) following the intuition of *steepest descending curves*, and they all coincide (e.g. [32], Prop 23.1, Rem 23.4) on Riemannian manifolds. Particularly, the GF defined by the MMS, which the PO method uses, is equivalent to the GF on Riemannian manifolds ([1], Thm 11.1.6; [7], Lemma 2.7). Now consider the Kullback–Leibler divergence (KLD) (or relative entropy in some literatures) that we want to minimize in this paper:

$$\mathrm{KL}(\cdot||\boldsymbol{p}) : \boldsymbol{q} \mapsto \begin{cases} \int_{\mathcal{X}} \log\left(\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}\boldsymbol{p}}\right)\mathrm{d}\boldsymbol{q}, & \text{if } \boldsymbol{q} \text{ is absolutely continuous (AC) wrt } \boldsymbol{p}, \\ +\infty, & \text{otherwise,} \end{cases}$$

where $\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}\boldsymbol{p}}$ is the Radon-Nikodym derivative. A curve $\boldsymbol{q}_t$ of the GF of the KLD is characterized by $\dot{\boldsymbol{q}}_t \sim v_t$ ([32], Thm 23.18; [1], Example 11.1.2) where $v_t = \nabla\log\left(\frac{\mathrm{d}\boldsymbol{q}_t}{\mathrm{d}\boldsymbol{p}}\right)$. Note that the GF of $\mathrm{KL}(\cdot||\boldsymbol{p})$ cannot be defined where $\boldsymbol{q}$ is not AC wrt $\boldsymbol{p}$. When $\mathrm{KL}(\cdot||\boldsymbol{p})$ is geodesically λ-convex on $\mathcal{P}_2$ (e.g. when p is λ-log-concave on $\mathcal{X}$ ([32], Thm 17.15)), $\boldsymbol{q}_t$ enjoys the exponential convergence $W_2(\boldsymbol{q}_t^{(1)},\boldsymbol{q}_t^{(2)}) \leq e^{-\lambda t}W_2(\boldsymbol{q}_0^{(1)},\boldsymbol{q}_0^{(2)})$ ([32], Thm 23.25, Thm 24.7; [1], Thm 11.1.4), as expected. We indistinguishably write $\mathrm{KL}(q||p)$ and $\mathbb{E}_q$ when the measures are AC.

2.3 Stein Variational Gradient Descent (SVGD)

SVGD [18] is a Bayesian inference method that updates particles by a proper vector field v so that the the distribution of particles q (assumed AC) approaches the posterior p (typically AAC) as fast as possible in terms of $\mathrm{KL}(q||p)$. Specifically, v is found by maximizing $-\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{KL}(q_t||p)\big|_{t=0} = \mathbb{E}_q[\nabla\log p\cdot v + \nabla\cdot v]$, where q_t with $q_0 = q$ is the varying density of moving particles with velocity v. By choosing $v \in \mathcal{H}^D$ where $\mathcal{H}$ is the reproducing kernel Hilbert space (RKHS) of a kernel $K(\cdot,\cdot)$, the optimal v can be solved explicitly: $v_{\mathcal{H}}(\cdot) = \mathbb{E}_{q(x)}[K(x,\cdot)\nabla\log p(x) + \nabla_x K(x,\cdot)]$. Samples are then updated using $v_{\mathcal{H}}$, which is estimated by averaging over the samples.

The SVGD vector field $v_{\mathcal{H}}$ is then interpreted as the tangent vector of the GF of the KLD on the $\mathcal{H}$-Wasserstein space $\mathcal{P}_{\mathcal{H}}$ [17], whose tangent space is taken as $T_{\boldsymbol{q}}\mathcal{P}_{\mathcal{H}} := \{\nabla\cdot(\boldsymbol{q}\phi) : \phi \in \mathcal{H}^D\}$ (weak derivative description)[3]. However, we note that this may not be a natural tangent space:

[3] Or the vector field description: the quotient space $\mathcal{H}^D/\simeq$ where $\phi_1 \simeq \phi_2 : \nabla\cdot(\boldsymbol{q}(\phi_1 - \phi_2)) = 0$.

Remark 3. *Lemma 1 guarantees that for* any *AC curve that passes $\boldsymbol{q}$, the unique existence of a tangent vector of the curve at $\boldsymbol{q}$ in $T_{\boldsymbol{q}}\mathcal{P}_2$ is guaranteed. But there is no such guarantee in $T_{\boldsymbol{q}}\mathcal{P}_{\mathcal{H}}$. Therefore tangent vectors in $T_{\boldsymbol{q}}\mathcal{P}_{\mathcal{H}}$ may not be able to exactly fit a GF on $\mathcal{P}_{\mathcal{H}}$ at $\boldsymbol{q}$.*

3 Bayesian Inference via Gradient Flow on $\mathcal{P}_2$

We aim to do Bayesian inference via simulating the gradient flow (GF) of $\mathrm{KL}(\cdot||\boldsymbol{p})$ on $\mathcal{P}_2$ which approaches to $\boldsymbol{p}$, the target posterior distribution. For typical inference tasks $\boldsymbol{p}$ is AC, thus is any measure on the GF since it should be AC wrt $\boldsymbol{p}$, as mentioned. Therefore, we can use density functions to write the vector field form of the GF of the KLD introduced in Section 2.2:

$$v_t(x) = \nabla \log p(x) - \nabla \log q_t(x).$$

The simulation of the GF follows the physical intuition detailed in Section 2.1, i.e. update samples $\{x_t^{(i)}\}_i$ of q_t by $x_{t+h}^{(i)} = x_t^{(i)} + hv_t(x^{(i)})$ then $\{x_{t+h}^{(i)}\}_i$ is a set of samples of q_{t+h}.

Remark 4. *We note that the deterministic dynamics $\mathrm{d}x = v_t(x)\mathrm{d}t$ with v_t above, produces the same evolution rule for the probability density q_t of x as the Langevin dynamics: $\mathrm{d}x = \nabla \log p(x)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t(x)$, where $B_t(x)$ is the Brownian motion, due to the Fokker-Planck equation. This meets the known recognition of the Langevin dynamics as the GF of the KLD on $\mathcal{P}_2$ (e.g. [12]).*

Now we engage in estimating v_t. We first analyze the requirements of two possible ways to do this and further the flexibility of PSVIs, then propose our methods GFSD/GFSF following the two ways. We fix an arbitrary time t and drop the subscript for notation brevity.

3.1 Smoothing Density or Smoothing Test Functions

The key obstacle to estimating v is the approximation of $-\nabla \log q(x)$. We know that $\{x^{(i)}\}_{i=1}^N$ is a finite set of samples of q, but we cannot directly approximate q by the empirical distribution[4] $\hat{q}(x) := \frac{1}{N}\sum_{i=1}^N \delta(x - x^{(i)})$ where δ is the Dirac delta function, since $\hat{q}$ is not AC thus the GF of the KLD is undefined at $\hat{q}$. To well-define the vector field, a straightforward way is to smooth $\hat{q}$ with a smooth kernel K on $\mathcal{X}$, leading to the AC approximation $\tilde{q}(x) := (\hat{q} * K)(x) = \frac{1}{N}\sum_{i=1}^N K(x, x^{(i)})$ (where "$*$" denotes convolution).

Beside the expression in density, we shall show in Section 3.2.2 that the vector field can also be characterized by the optimization problem of the form $\min_{\varphi \in \mathcal{C}_c^\infty(\mathcal{X})} \mathbb{E}_q[\varphi]$. By noting the equality $\mathbb{E}_{\tilde{q}}[\varphi] = \mathbb{E}_{\hat{q}*K}[\varphi] = \mathbb{E}_{\hat{q}}[\varphi * K]$ (due to the interchangeability of integral and summation), we claim that smoothing density is equivalent to smoothing test function in this optimization formulation. Furthermore, we show that when K is a Gaussian kernel, smoothing test functions in $\mathcal{C}_c^\infty(\mathcal{X})$ is equivalent to taking test functions from the RKHS of K.

Proposition 5. *For Gaussian kernel K and $\mathcal{X} = \mathbb{R}^D$, $\mathcal{G} := \overline{\{\varphi * K : \varphi \in \mathcal{C}_c^\infty(\mathcal{X})\}}^{L^2(\mathcal{X})}$ is isometrically isomorphism to the RKHS $\mathcal{H}$ of K.*

Proof is provided in Appendix A1. As we shall analyze below, other PSVIs also have to face this issue. We claim that for all PSVIs, either density or test function has to be a smoothed one, and PSVIs have this extend of flexibility. This is an intrinsic requirement by the KLD that they minimize.

Case study: SVGD As introduced in Section 2.3, SVGD identifies its vector field by solving

$$\max_{v \in L^2(p;\mathbb{R}^D), \|v\|=1} \mathbb{E}_q[\nabla \log p \cdot v + \nabla \cdot v], \tag{1}$$

where $v \in L^2(p;\mathbb{R}^D) \supset \mathcal{C}_c^\infty(\mathcal{X};\mathbb{R}^D)$ is required by the condition of Stein's identity [17]. We first claim that neither smoothing density q nor smoothing test function v will disable the problem to identify the vector field, as the GF of $\mathrm{KL}(\cdot||p)$ is not defined at $\hat{q}$.

Proposition 6. *For $q = \hat{q}$ and $v \in L^2(p;\mathbb{R}^D)$, problem (1) has no optimal solution. In fact the supremum of the objective is infinite, indicating that a maximizing sequence of v tend to be ill-posed.*

Proof is provided in Appendix A2. Intuitively, without a restriction on the sharpness of the test functions, v could be extremely peaked around the samples, which produces unreasonably large

[4] $\hat{\boldsymbol{q}}$ is not AC and rigorously $\hat{q}$ is not a density function. We informally treat it as a special function to avoid verbose language.

vectors at the samples. If we choose to smooth the test function v with a Gaussian kernel K, we only need to choose v from $\mathcal{H}^D$ according to Proposition 5, which is exactly what SVGD does. This in turn is equivalent to smoothing the density by noticing $\mathbb{E}_{\hat{q}}[\nabla \log p \cdot (v * K) + \nabla \cdot (v * K)] = \mathbb{E}_{\hat{q}}[(\nabla \log p \cdot v) * K + (\nabla \cdot v) * K] = \mathbb{E}_{(\hat{q}*K)}[\nabla \log p \cdot v + \nabla \cdot v]$.

SVGD makes no assumption on the form of the density q as long as its samples are known, but it actually transmits the restriction on q to the test function v. The choice for v in $\mathcal{H}^D$ is not just for a tractable solution, but more importantly, for guaranteeing a valid vector field. There is no free lunch in the approximation flexibility. This claim also holds for the particle optimization (PO) method [3], since it adopts the vector field of SVGD for the KLD term in its optimization objective in each step.

3.2 Simulating the Gradient Flow on $\mathcal{P}_2$

Now we propose two practical methods based on smoothing density and smoothing test functions for simulating the gradient flow (GF) on $\mathcal{P}_2$ as Bayesian inference methods. A novel bandwidth selection method is also proposed.

3.2.1 Gradient Flow with Smoothed Density (GFSD)

By directly estimating q with the smoothed density $\tilde{q} = \hat{q} * K$, we compute the vector field as $v(t) = \nabla \log p(x) - \nabla \log \tilde{q}(x) = \nabla \log p(x) - \big(\sum_i \nabla_x K(x, x^{(i)})\big) / \big(\sum_j K(x, x^{(j)})\big)$. We call this method the Gradient Flow with Smoothed Density (GFSD).

We note that GFSD is closely related to SVGD: by the method of variation, the optimal solution to problem (1) with $q = \tilde{q}$ and $v \in L^2(\tilde{q}; \mathbb{R}^D)$ is proportional to the GFSD vector field. This indicates that the smoothing density version of SVGD coincides with GFSD. In fact, the revised optimization problem characterizes the gradient of the KLD on $\mathcal{P}_2$.

3.2.2 Gradient Flow with Smoothed Test Functions (GFSF)

For the problematic part $u(x) = -\nabla \log q(x)$, or equivalently $q(x)u(x) + \nabla q(x) = 0$, we treat it as an equality that holds in the weak sense, which means $\mathbb{E}_q[\phi \cdot u - \nabla \cdot \phi] = 0, \forall \phi \in \mathcal{C}_c^\infty(\mathcal{X}; \mathbb{R}^D)$[5]. We take $q = \hat{q}$ and smooth the test function, which means taking ϕ from $\mathcal{H}^D$ according to Proposition 5. To enforce the equality to hold, we require the desired u to solve the following optimization problem:

$$\min_u \max_{\phi \in \mathcal{H}^D, \|\phi\|=1} \big(\mathbb{E}_{\hat{q}}[\phi \cdot u - \nabla \cdot \phi]\big)^2 = \frac{1}{N^2}\Big(\sum_{i=1}^N \big(\phi(x^{(i)}) \cdot u(x^{(i)}) - \nabla \cdot \phi(x^{(i)})\big)\Big)^2. \quad (2)$$

The closed-form solution of the problem is $\hat{u} = \hat{K}' \hat{K}^{-1}$ in matrix form, where $\hat{u}_{:,i} = u(x^{(i)})$, $\hat{K}_{ij} = K(x^{(i)}, x^{(j)})$, and $\hat{K}'_{:,i} = \sum_j \nabla_{x^{(j)}} K(x^{(j)}, x^{(i)})$ (see Appendix A3). So the total vector field in matrix form can be expressed as $g + \hat{K}' \hat{K}^{-1}$, where $g_{:,i} = \nabla_{x^{(i)}} \log p(x^{(i)})$. We call this method the Gradient Flow with Smoothed test Functions (GFSF). Note that the inverse $\hat{K}^{-1}$ does not incur a serious computational cost since PSVIs are particle-efficient and we do not need a large particle size.

An interesting relation of GFSF to SVGD is that the SVGD vector field can be expressed as $\hat{K}g + \hat{K}'$. We also note that the GFSF estimate of $-\nabla \log q$ coincides with the method of [15], which is derived by using Stein's identity (a more general form of the weak derivative) and choosing one particular test function. In the work, gradient in the data space is estimated to train implicit generative models.

For both GFSD and GFSF, the only information needed on p is the gradient $\nabla \log p$, which is tractable for Bayesian inference tasks. Unlike SVGD, the gradient term in both our methods is not weighted-averaged among all particles, thus each particle moves more efficiently towards the high-probability region around it. Moreover, due to the relation to the Langevin dynamics (LD) (Remark 4), both methods can adopt stochastic gradient for large scale inference tasks, in the same way that LD does [35].

3.2.3 Bandwidth Selection via Heat Equation

Since kernel is involved to smooth either density or test functions, the bandwidth of the kernel will affect the estimation of the vector field. SVGD chooses the bandwidth with a median method, which is based on a numerical consideration. Here we propose a novel method for selecting the bandwidth by exploring the dynamics of the gradient flow.

[5] We also consider scalar-valued test functions $\varphi \in \mathcal{C}_c^\infty(\mathcal{X})$ and smooth them in $\mathcal{H}$, which gives the same result, as shown in Appendix A4.

As noted in Remark 4, the deterministic dynamics of GF and the Langevin dynamics realizes the same rule of density evolution. Particularly, both the deterministic dynamics $\mathrm{d}x = -\nabla \log q_t(x)\mathrm{d}t$ and the Brownian motion $\mathrm{d}x = \sqrt{2}\mathrm{d}B_t(x)$ produce the evolution rule of the heat equation (HE) $\partial_t q_t(x) = \Delta q_t(x)$. So we would like to select the bandwidth of the smoothing kernel by enforcing the vector field $-\nabla \log \tilde{q}$ to recover the effect of HE. Specifically, we explicitly express the dependence of $\tilde{q}(x)$ on the samples as $\tilde{q}(x; \{x^{(j)}\}_j)$ when needed, which is an approximation on the current density q_t. After an infinitesimal time ε, $q_{t+\varepsilon}(x)$ should be approximated by both $q_t + \varepsilon\Delta q_t \approx \tilde{q} + \varepsilon\Delta\tilde{q}$ due to the HE, and by $\tilde{q}_{t+\varepsilon} = \tilde{q}(x; \{x^{(i)} - \varepsilon\nabla \log \tilde{q}(x^{(i)})\}_i)$ due to the deterministic dynamics. As a result, we require the equality $\Delta\tilde{q}(x; \{x^{(i)}\}_i) + \sum_j \nabla_{x^{(j)}}\tilde{q}(x; \{x^{(i)}\}_i) \cdot \nabla \log \tilde{q}(x^{(j)}; \{x^{(i)}\}_i) = 0$ to hold. So in practice, a reasonable bandwidth can be selected by minimizing

$$\sum_k \Big(\Delta\tilde{q}(x^{(k)}; \{x^{(i)}\}_i) + \sum_j \nabla_{x^{(j)}}\tilde{q}(x^{(k)}; \{x^{(i)}\}_i) \cdot \nabla \log \tilde{q}(x^{(j)}; \{x^{(i)}\}_i)\Big)^2. \tag{3}$$

We call it the HE method. Due to the equivalence of smoothing density and smoothing test functions analyzed in Section 3.1, the HE method can also be applied to methods based on smoothing test functions, e.g. SVGD and GFSF. Implementation details are provided in Appendix A5.

4 Accelerated First-order Methods on $\mathcal{P}_2$

The gradient flow methods described in the above section can be regarded as the gradient descent method on Riemannian manifold. Beyond that, there exist accelerated first-order methods on Riemannian manifold, e.g. Riemannian Nesterov's Accelerated Gradient (RNAG) [20], that enjoy a faster convergence rate. To apply the RNAG method to the Riemannian manifold $\mathcal{P}_2$, two more ingredients are required: exponential map (and its inverse) and parallel transport. Intuitively, the exponential map $\mathrm{Exp}_{\boldsymbol{q}} : T_{\boldsymbol{q}}\mathcal{P}_2 \to \mathcal{P}_2$ acts as the addition (displacement) of the point $\boldsymbol{q}$ with a vector to get to another point, and the parallel transport $\Gamma_{\boldsymbol{q_1}}^{\boldsymbol{q_2}} : T_{\boldsymbol{q_1}}\mathcal{P}_2 \to T_{\boldsymbol{q_2}}\mathcal{P}_2$ relates tangent vectors at different points and should be invoked before using a tangent vector at a different point.

Exponential map on $\mathcal{P}_2$ As indicated by Corollary 7.22 and Theorem 10.38 of [32], for an AC measure $\boldsymbol{q}$ and $\xi \in T_{\boldsymbol{q}}\mathcal{P}_2$, $\mathrm{Exp}_{\boldsymbol{q}}\xi = (\mathrm{id} + \xi)_{\#}\boldsymbol{q}$ (notation defined in Lemma 2). For its inverse, consider two AC measures $\boldsymbol{q_1}, \boldsymbol{q_2}$ that are close. Let $\{x_1^{(j)}\}_j$ be a set of samples of $\boldsymbol{q_1}$ and $\{x_2^{(j)}\}_j$ of $\boldsymbol{q_2}$, and assume that they are pairwisely close: $\|x_1^{(j)} - x_2^{(j)}\| \ll \min\{\min_i \|x_1^{(i)} - x_1^{(j)}\|, \min_i \|x_2^{(i)} - x_2^{(j)}\|\}, \forall j$. Then the optimal transport map from $\boldsymbol{q_1}$ to $\boldsymbol{q_2}$ satisfies $\mathcal{T}_{\boldsymbol{q_1}}^{\boldsymbol{q_2}}(x_1^{(j)}) = x_2^{(j)}, \forall j$. According to Theorem 7.2.2 of [1], we have $\mathcal{T}_{\boldsymbol{q_1}}^{\boldsymbol{q_t}} = (1-t)\mathrm{id} + t\mathcal{T}_{\boldsymbol{q_1}}^{\boldsymbol{q_2}}$ where $(\boldsymbol{q}_t)_{t\in[0,1]}$ is the geodesic (in the Riemannian manifold sense) from $\boldsymbol{q_1}$ to $\boldsymbol{q_2}$, and Proposition 8.4.6 further asserts that $\dot{\boldsymbol{q}}_0 \sim \lim_{t\to 0}\frac{1}{t}(\mathcal{T}_{\boldsymbol{q_1}}^{\boldsymbol{q_t}} - \mathrm{id}) = \mathcal{T}_{\boldsymbol{q_1}}^{\boldsymbol{q_2}} - \mathrm{id}$ (vector field form), while by definition $\mathrm{Exp}_{\boldsymbol{q_1}}^{-1}(\boldsymbol{q_2}) \sim \dot{\boldsymbol{q}}_0$. So we have $\big(\mathrm{Exp}_{\boldsymbol{q_1}}^{-1}(\boldsymbol{q_2})\big)(x_1^{(j)}) = x_2^{(j)} - x_1^{(j)}, \forall j$. Note that only the knowledge on the samples is sufficient for our use to update the samples. Also note that when we apply $\mathrm{Exp}_{\cdot}^{-1}(\cdot)$ in the following, the two measures are close since one is derived by an infinitesimal displacement of another, so is each pair of their samples $x_1^{(j)}$ and $x_2^{(j)}$ (see Appendix A6.1, A6.2).

Parallel Transport on $\mathcal{P}_2$ We first note that there have been formal researches on the parallel transport on $\mathcal{P}_2$ [21, 22], but the result requires differentiating the vector field thus hard to implement.

We propose to use the Schild's ladder method [6, 13], which provides a tractable first-order approximation of the parallel transport $\Gamma_{\boldsymbol{q_1}}^{\boldsymbol{q_2}}$. As shown in Fig. 1, given $\boldsymbol{q_1}$, $\boldsymbol{q_2}$ and $\xi \in T_{\boldsymbol{q_1}}\mathcal{P}_2$, the procedure to approximate $\tilde{\Gamma}_{\boldsymbol{q_1}}^{\boldsymbol{q_2}}\xi$ is (i) find the point $\mathrm{Exp}_{\boldsymbol{q_1}}\xi$; (ii) find the midpoint of the geodesic from $\boldsymbol{q_2}$ to $\mathrm{Exp}_{\boldsymbol{q_1}}\xi$: $\boldsymbol{q_M} = \mathrm{Exp}_{\boldsymbol{q_1}}(\frac{1}{2}\mathrm{Exp}_{\boldsymbol{q_1}}^{-1}(\mathrm{Exp}_{\boldsymbol{q_1}}\xi))$; (iii) extrapolate the geodesic from $\boldsymbol{q_1}$ to $\boldsymbol{q_M}$ by doubling the length to find $\boldsymbol{q_E} := \mathrm{Exp}_{\boldsymbol{q_1}}(2\mathrm{Exp}_{\boldsymbol{q_1}}^{-1}(\boldsymbol{q_M}))$; (iv) the approximator is taken as $\tilde{\Gamma}_{\boldsymbol{q_1}}^{\boldsymbol{q_2}}\xi = \mathrm{Exp}_{\boldsymbol{q_2}}^{-1}(\boldsymbol{q_E})$. Note that the Schild's ladder method only requires the exponential map and its inverse.

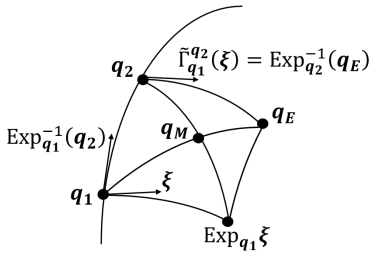


Figure 1: Illustration of the Schild's ladder method. Figure inspired by [13].

Following the procedure, we find that for AC measures $\boldsymbol{q_1}, \boldsymbol{q_2}$ with pairwisely close samples $\{x_1^{(j)}\}_j, \{x_2^{(j)}\}_j$, the approximation to the parallel transport satisfies $\big(\tilde{\Gamma}_{\boldsymbol{q_1}}^{\boldsymbol{q_2}}(\xi)\big)(x_2^{(j)}) = \xi(x_1^{(j)})$ (See Appendix A6.1). By applying the RNAG algorithm [20] on $\mathcal{P}_2$ (details in Appendix A6.2), we

have the Wasserstein Nesterov's Accelerated Gradient method (WNAG), as presented in Alg. 1. To highlight the difference, we call the vanilla implementation of PSVIs as Wasserstein Gradient Descent method (WGD). We emphasize that one cannot directly apply the vanilla Nesterov's acceleration method [24] on the update rule of every single particle, since each particle is not optimizing any function. Although it needs more investigation to see whether the algorithm suits for SVGD, we can apply it algorithmically, and it also makes a salient improvement in experiments.

The tools developed here also make it possible to apply other optimization techniques on Riemannian manifold to benefit PSVIs, e.g. Riemannian BFGS [8, 28, 36] and Riemannian stochastic variance reduction gradient [37]. We leave these extensions as future work.

Algorithm 1 Wasserstein Nesterov's Acceleration Gradient method (WNAG)

Select acceleration factor $\alpha > 3$; Initialize $\{x_0^{(j)}\}_{j=1}^N$ with distinct values; let $y_0^{(j)} = x_0^{(j)}$;
for $k = 1, 2, \cdots$, **do**
  Determine bandwidth using either the median or the HE method for $\{y_{k-1}^{(j)}\}_{j=1}^N$;
  for $i = 1, \cdots, N$, **do**
    Find the value $\xi_{k-1}^{(i)}$ of the vector field at $y_{k-1}^{(i)}$ by SVGD/GFSD/GFSF;
    $x_k^{(i)} = y_{k-1}^{(i)} + \varepsilon \xi_{k-1}^{(i)}$;
    $y_k^{(i)} = x_k^{(i)} + \frac{k-1}{k}(y_{k-1}^{(i)} - x_{k-1}^{(i)}) + \frac{k+\alpha-2}{k}\varepsilon \xi_{k-1}^{(i)}$.
  end for
end for

5 Experiments[6]

5.1 Toy Experiments

We first investigate the validity of GFSD and GFSF as well as the benefit of the HE method by showing 200 samples that they produce for a toy bimodal distribution (following [29]), and compare with SVGD, and the particle optimization method (PO) [3] as an independent PSVI method. As shown in Fig. 2, when using the median method for bandwidth, samples of GFSD and GFSF tend to collapse, since the median method cannot make $\nabla \log q$ act the same as the Brownian motion, which is responsible for diversity. With the HE method for bandwidth, GFSD and GFSF produce well-aligned samples with a reasonable diversity. Amazingly, we can see that the samples are uniformly distributed along the contour of the target density, indicating that they form a better set of samples to represent the density. We also note that the PO method does not improve the ultimate sample distribution of SVGD, while our HE method can also make samples of SVGD and PO align more neatly.

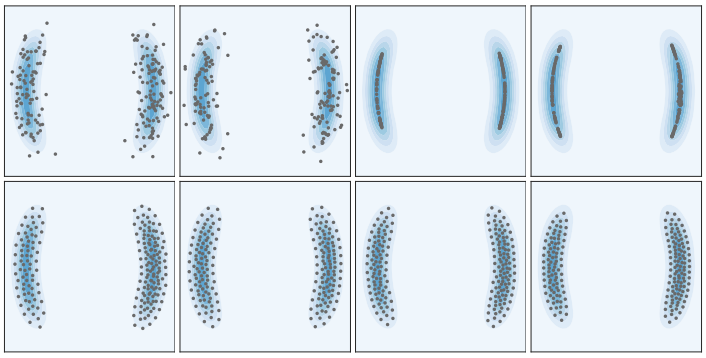
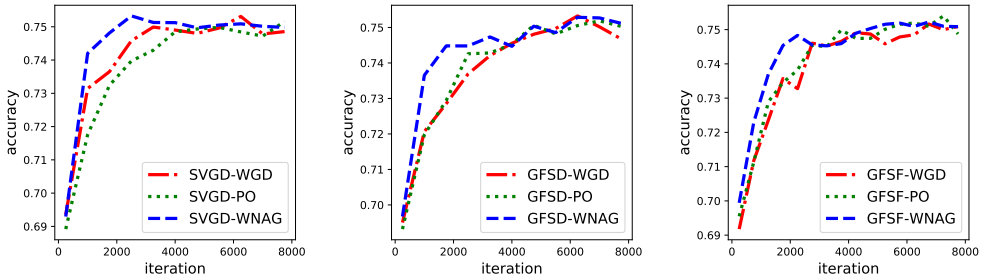


Figure 2: The validity of GFSD/GFSF and the effect of the HE method. Samples are plotted as grey dots and the common target distribution as blue shade. The columns correspond to SVGD, PO, GFSD, GFSF methods respectively, and the rows correspond to the median and HE method for kernel bandwidth. All methods are run for 400 iterations with the same initialization.

5.2 Bayesian Logistic Regression

We conduct the standard Bayesian logistic regression experiment on the Covertype dataset, following the same settings as [18], except we average the results over 10 random trials. The SVGD-WGD

[6]Codes and data available at `http://ml.cs.tsinghua.edu.cn/~changliu/awgf/`



(a) SVGD (b) GFSD (c) GFSF

Figure 3: Acceleration effect of WNAG over WGD with comparison to PO on the Bayesian logistic regression on the Covertype dataset. Each curve is averaged over 10 runs with random 80%train-20%test split of the dataset and the mini-batch size is taken as 50, following [18].

method uses AdaGrad with momentum for adjusted step size so that it is identical to the vanilla SVGD method. Other methods use a shrinking step size scheme in the face of stochastic gradient. We select parameters for all methods for the fastest convergence with a stable result. Although the PO method does not strictly count for an acceleration method for PSVIs, we treat it as an empirical acceleration implementation here.

We first find that both the median method and the HE method for bandwidth selection achieve similar results in all cases, and we skip showing the comparison. We then show the improvement of iteration convergence rate by the WNAG acceleration over WGD and the comparison with the PO acceleration in Fig. 3. It is shown that WNAG gains a salient speed-up over WGD for all of SVGD, GFSD and GFSF methods, and it outperforms the PO method which does not make a significant improvement over WGD, as also shown in [3]. We also note that GFSD and GFSF methods achieve a comparable result as SVGD, indicating their validity.

5.3 Bayesian Neural Network

We test all the methods on the Bayesian neural network, following the settings described in [18], except we run all methods for the same amount of iterations. For the SVGD-WGD method, we take either the reported result in [18] or the result we observe, whichever is better. Table 1 presents the results on one of the datasets, Kin8nm, that Liu *et al.* [18] use, and Appendix A7 shows more results on other datasets. We observe that the WNAG acceleration method saliently outperforms the WGD and PO methods for all of SVGD, GFSD and GFSF on all datasets. The PO method, as an empirical acceleration method, also improves the performance, but not to the extent of the WNAG method. The GFSD and GFSF methods also achieve better performance than SVGD in most cases.

Table 1: Results on Bayesian neural network on the Kin8nm dataset. Results are averaged over 20 runs with random 90%train-10%test split and the mini-batch size is taken as 100, following [18].

Method	Avg. Test RMSE			Avg. Test LL		
	SVGD	GFSD (Ours)	GFSF (Ours)	SVGD	GFSD (Ours)	GFSF (Ours)
WGD	0.084±0.002	0.080±0.003	0.083±0.002	1.042±0.016	1.087±0.029	1.044±0.016
PO	0.078±0.002	0.081±0.002	0.080±0.002	1.114±0.022	1.067±0.017	1.073±0.016
WNAG (Ours)	0.070±0.002	0.071±0.001	**0.070±0.001**	1.167±0.015	1.167±0.017	**1.190±0.014**

6 Conclusion

We consider doing Bayesian inference by minimizing the KLD on the 2-Wasserstein space $\mathcal{P}_2$, whose Riemannian structure enables us to do develop various methods. We analyze the flexibility of such particle-simulation-based variational inference methods (PSVIs) of up to a smoothed density or a smoothed test function family, and develop two PSVIs by simulating the gradient flow on $\mathcal{P}_2$ by smoothing density and test functions, respectively, with a principled kernel selection method HE. We also propose an acceleration method WNAG for PSVIs by exploring the Riemannian structure of $\mathcal{P}_2$. Experiments show the validity of GFSD/GFSF with the HE method, and the improved iteration convergence rate of the WNAG method for all PSVIs.

References

[1] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

[2] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

[3] Changyou Chen and Ruiyi Zhang. Particle optimization in stochastic gradient mcmc. *arXiv preprint arXiv:1711.10927*, 2017.

[4] Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. In *Advances in neural information processing systems*, pages 3203–3211, 2014.

[5] Manfredo Perdigao Do Carmo. *Riemannian Geometry*. 1992.

[6] J Ehlers, F Pirani, and A Schild. The geometry of free fall and light propagation, in the book "general relativity"(papers in honour of jl synge), 63–84, 1972.

[7] Matthias Erbar et al. The heat equation on manifolds as a gradient flow in the wasserstein space. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 46, pages 1–23. Institut Henri Poincaré, 2010.

[8] Daniel Gabay. Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, 37(2):177–219, 1982.

[9] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. In *Readings in Computer Vision*, pages 564–584. Elsevier, 1987.

[10] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165*, 2017.

[11] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[12] Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

[13] Arkady Kheyfets, Warner A Miller, and Gregory A Newton. Schild's ladder parallel transport procedure for an arbitrary connection. *International Journal of theoretical Physics*, 39(12):2891–2898, 2000.

[14] Ondrej Kováčik and Jiří Rákosník. On spaces $l\hat{p}(x)$ and $w\hat{k}$, p(x). *Czechoslovak Mathematical Journal*, 41(4):592–618, 1991.

[15] Yingzhen Li and Richard E Turner. Gradient estimators for implicit models. *arXiv preprint arXiv:1705.07107*, 2017.

[16] Chang Liu and Jun Zhu. Riemannian stein variational gradient descent for bayesian inference. *arXiv preprint arXiv:1711.11216*, 2017.

[17] Qiang Liu. Stein variational gradient descent as gradient flow. In *Advances in neural information processing systems*, pages 3118–3126, 2017.

[18] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pages 2378–2386, 2016.

[19] Yang Liu, Prajit Ramachandran, Qiang Liu, and Jian Peng. Stein variational policy gradient. *arXiv preprint arXiv:1704.02399*, 2017.

[20] Yuanyuan Liu, Fanhua Shang, James Cheng, Hong Cheng, and Licheng Jiao. Accelerated first-order methods for geodesically convex optimization on riemannian manifolds. In *Advances in Neural Information Processing Systems*, pages 4875–4884, 2017.

[21] John Lott. Some geometric calculations on wasserstein space. *Communications in Mathematical Physics*, 277(2):423–437, 2008.

[22] John Lott. An intrinsic parallel transport in wasserstein space. *Proceedings of the American Mathematical Society*, 145(12):5329–5340, 2017.

[23] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11), 2011.

[24] Yurii Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

[25] Felix Otto. The geometry of dissipative evolution equations: the porous medium equation. 2001.

[26] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[27] Yunchen Pu, Zhe Gan, Ricardo Henao, Chunyuan Li, Shaobo Han, and Lawrence Carin. Vae learning via stein variational gradient descent. In *Advances in Neural Information Processing Systems*, pages 4239–4248, 2017.

[28] Chunhong Qi, Kyle A Gallivan, and P-A Absil. Riemannian bfgs algorithm with applications. In *Recent advances in optimization and its applications in engineering*, pages 183–192. Springer, 2010.

[29] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

[30] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[31] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[32] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[33] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

[34] Dilin Wang and Qiang Liu. Learning to draw samples: With application to amortized mle for generative adversarial learning. *arXiv preprint arXiv:1611.01722*, 2016.

[35] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

[36] Xinru Yuan, Wen Huang, P-A Absil, and Kyle A Gallivan. A riemannian limited-memory bfgs algorithm for computing the matrix geometric mean. *Procedia Computer Science*, 80:2147–2157, 2016.

[37] Hongyi Zhang, Sashank J Reddi, and Suvrit Sra. Riemannian svrg: fast stochastic optimization on riemannian manifolds. In *Advances in Neural Information Processing Systems*, pages 4592–4600, 2016.

[38] Ding-Xuan Zhou. Derivative reproducing properties for kernel methods in learning theory. *Journal of computational and Applied Mathematics*, 220(1):456–463, 2008.

[39] Jingwei Zhuo, Chang Liu, Jiaxin Shi, Jun Zhu, Ning Chen, and Bo Zhang. Message passing stein variational gradient descent. *arXiv preprint arXiv:1711.04425*, 2017.

Appendix

A1: Proof of Proposition 5

Proof. Note that $\overline{\mathcal{C}_c^\infty(\mathbb{R}^D)}^{L^2(\mathbb{R}^D)} = L^2(\mathbb{R}^D)$ (e.g. [14], Thm 2.11), and the map $\varphi \mapsto \varphi * K, L^2(\mathbb{R}^D) \to L^2(\mathbb{R}^D)$ is continuous. So we have $\mathcal{G} = \{\varphi * K : \varphi \in L^2(\mathbb{R}^D)\}$. On the other hand, due to Proposition 4.46 and Theorem 4.47 of [30], the map $\varphi \mapsto \varphi * K$ is an isometric isomorphism between $\mathcal{G}$ and $\mathcal{H}$, the reproducing kernel Hilbert space of K. This completes the proof. □

A2: Proof of Proposition 6

Proof. To show that the problem below has no solution,

$$\sup_{v\in L^2(p;\mathbb{R}^D),\|v\|=1} \sum_{i=1}^N \Big(\nabla \log p(x^{(i)}) \cdot v(x^{(i)}) + \nabla \cdot v(x^{(i)})\Big), \tag{4}$$

we will find a sequence of functions $\{v_n\}$ satisfying conditions in (4) while the objective goes to infinity.

We assume that there exists $r_0 > 0$ such that $p(x) > 0$ for any $\|x - x^{(i)}\|_\infty < r_0$, $i = 1, 2, \ldots, N$, which is reasonable because it is almost impossible to sample $x^{(i)}$ with $p(x)$ vanishes in every neighborhood of $x^{(i)}$.

Denoting $v(x) = (v_1(x), \ldots, v_D(x))^T$ for any D-dimensional vector function v and $\nabla f(x) = (\partial_1 f(x), \ldots, \partial_D f(x))^T$ for any real-valued function f, the objective can be written as

$$\begin{aligned}
\mathcal{L}_v &= \sum_{i=1}^N \Big(\nabla \log p(x^{(i)}) \cdot v(x^{(i)}) + \nabla \cdot v(x^{(i)})\Big) \\
&= \sum_{i=1}^N \Big(\sum_{\alpha=1}^D \partial_\alpha[\log p(x^{(i)})] v_\alpha(x^{(i)}) + \sum_{\alpha=1}^D \partial_\alpha[v_\alpha(x^{(i)})]\Big) \\
&= \sum_{\alpha=1}^D \sum_{i=1}^N \Big(\partial_\alpha[\log p(x^{(i)})] v_\alpha(x^{(i)}) + \partial_\alpha[v_\alpha(x^{(i)})]\Big).
\end{aligned} \tag{5}$$

For every $v \in L^2(p;\mathbb{R}^D), \|v\| = 1$, we can define a function $\phi = (\phi_1, \ldots, \phi_D)^T \in L^2(\mathbb{R}^D)$ correspondingly, such that $\phi(x) = p(x)^{\frac{1}{2}} v(x)$, which means $\phi_\alpha(x) = p(x)^{\frac{1}{2}} v_\alpha(x)$ and

$$\begin{aligned}
\|\phi\|_2^2 &= \int_{\mathbb{R}^D} \phi^2 \mathrm{d}x = \int_{\mathbb{R}^D} \sum_{\alpha=1}^D (\phi_\alpha(x))^2 \mathrm{d}x \\
&= \int_{\mathbb{R}^D} \sum_{\alpha=1}^D (v_\alpha(x))^2 p(x) \mathrm{d}x = \|v\|^2 = 1.
\end{aligned}$$

Rewrite (5) in term of ϕ,

$$\begin{aligned}
\mathcal{L}_\phi &= \sum_{\alpha=1}^D \sum_{i=1}^N \Big(\partial_\alpha[\log p(x^{(i)})] v_\alpha(x^{(i)}) + \partial_\alpha[v_\alpha(x^{(i)})]\Big) \\
&= \sum_{\alpha=1}^D \sum_{i=1}^N \Big(\partial_\alpha[\log p(x^{(i)})] \phi_\alpha(x^{(i)}) p(x^{(i)})^{-\frac{1}{2}} + \partial_\alpha[\phi_\alpha(x^{(i)}) p(x^{(i)})^{-\frac{1}{2}}]\Big) \\
&= \sum_{\alpha=1}^D \sum_{i=1}^N \Big(\frac{1}{2} p(x^{(i)})^{-\frac{3}{2}} \partial_\alpha[p(x^{(i)})] \phi_\alpha(x^{(i)}) + p(x^{(i)})^{-\frac{1}{2}} \partial_\alpha[\phi_\alpha(x^{(i)})]\Big) \\
&= \sum_{\alpha=1}^D \sum_{i=1}^N \Big(A_\alpha^{(i)} \phi_\alpha(x^{(i)}) + B^{(i)} \partial_\alpha[\phi_\alpha(x^{(i)})]\Big),
\end{aligned} \tag{6}$$

where $A_\alpha^{(i)} = \frac{1}{2}p(x^{(i)})^{-\frac{3}{2}}\partial_\alpha[p(x^{(i)})]$ and $B^{(i)} = p(x^{(i)})^{-\frac{1}{2}} > 0$. We will now construct a sequence $\{\phi_n\}$ to show the problem

$$\inf_{\phi\in L^2(\mathbb{R}^D),\|\phi\|=1} \sum_{\alpha=1}^{D}\sum_{i=1}^{N}\left(A_\alpha^{(i)}\phi_\alpha(x^{(i)}) + B^{(i)}\partial_\alpha[\phi_\alpha(x^{(i)})]\right) \tag{7}$$

has no solution, then induce a sequence $\{v_n\}$ by $\{\phi_n\}$ for problem (4).

Define a sequence of functions

$$\chi_n(x) = \begin{cases} I_n^{-1/2}(1-x^2)^{n/2}, \text{ for } x \in [-1,1], \\ 0, \text{ otherwise.} \end{cases}$$

We have $\int_{\mathbb{R}} \chi_n(x)^2 \mathrm{d}x = 1$ with $I_n = \int_{-1}^{1}(1-x^2)^n \mathrm{d}x = \sqrt{\pi}\frac{\Gamma(n+1)}{\Gamma(n+3/2)}$, where $\Gamma(\cdot)$ is the Gamma function. Note that when $x = -1/\sqrt{n}$,

$$\begin{aligned} \chi_n'(x) =& -nI_n^{-\frac{1}{2}}x(1-x^2)^{\frac{n-2}{2}} \\ =& \pi^{-\frac{1}{4}}\sqrt{\frac{\Gamma(n+\frac{3}{2})}{\Gamma(n+1)}}\sqrt{n}(1-\frac{1}{n})^{\frac{n-2}{2}} && (x = -\frac{1}{\sqrt{n}}) \\ >& \pi^{-\frac{1}{4}}\sqrt{n}(1-\frac{1}{n})^{\frac{n-2}{2}}, && (\Gamma(n+\frac{3}{2}) > \Gamma(n+1)) \end{aligned} \tag{8}$$

therefore,

$$\lim_{n\to\infty}\chi_n'(-\frac{1}{\sqrt{n}}) > \lim_{n\to\infty}\pi^{-\frac{1}{4}}\sqrt{n}(1-\frac{1}{n})^{\frac{n-2}{2}} = \pi^{-\frac{1}{4}}e^{-\frac{1}{2}}\lim_{n\to\infty}\sqrt{n} = +\infty.$$

Denote $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_D^{(i)})^T \in \mathbb{R}^D, i = 1, \ldots, N$ and

$$r_1 = \frac{1}{3}\min_{i\neq j}\|x^{(i)} - x^{(j)}\|_\infty = \frac{1}{3}\min_{\alpha\in\{1,\ldots,D\}, i\neq j}|x_\alpha^{(i)} - x_\alpha^{(j)}|.$$

We extend χ_n to $\mathbb{R}^D$ as ξ_n with support $\mathrm{supp}(\xi_n) = [-r, r]^D$,

$$\xi_n(x_1, x_2, \ldots, x_D) = r^{-D/2}\prod_{\alpha=1}^{D}\chi_n(\frac{x_i}{r}), \tag{9}$$

where $r = \min\{r_0, r_1\}$. It is easy to show that $\int_{\mathbb{R}^D}\xi_n(x)^2\mathrm{d}x = 1$, and

$$\lim_{n\to\infty}\partial_\alpha\xi_n(-\epsilon_n) = +\infty, \ \alpha = 1, 2, \ldots, D, \tag{10}$$

with $\epsilon_n = \frac{r}{\sqrt{n}}(1, 1, \ldots, 1)^T$.

We choose $\phi_\alpha(x) = \frac{1}{ND}\sum_{i=1}^{N}\psi_\alpha^{(i)}$, where $\psi_\alpha^{(i)}$ is defined by

$$\psi_\alpha^{(i)}(x) = \begin{cases} \xi_n(x - x^{(i)} - \epsilon_n) \text{ if } A_\alpha^{(i)} >= 0 \\ -\xi_n(x - x^{(i)} + \epsilon_n) \text{ if } A_\alpha^{(i)} < 0 \end{cases} \tag{11}$$

With $\int_{\mathbb{R}^D}\psi_\alpha^{(i)}(x)\psi_\alpha^{(j)}(x)\mathrm{d}x = 0, \forall i \neq j$, we know ϕ_n satisfies conditions in (7). Note that $\forall i, j$, $A_\alpha^{(i)}\psi_\alpha^{(j)}(x^{(i)}) \geq 0$, and

$$\partial_\alpha\psi_\alpha^{(j)}(x^{(i)}) = \begin{cases} +\infty, \text{when } n\to\infty, \text{ if } i = j \\ 0, \text{ if } i \neq j, \end{cases}$$

we can see $\mathcal{L}_{\phi_n} \to +\infty$ in (6) when $n \to \infty$.

Since $\mathrm{supp}(\phi_n) \subset \mathrm{supp}(p)$, we can induce a sequence of $\{v_n\}$ from $\{\phi_n\}$ as $v_n = \phi_n/\sqrt{p(x)}$, which satisfies restrictions in (4) and the objective $\mathcal{L}_{v_n}$ will go to infinity when $n \to \infty$. Note that any element in $L^2(p; \mathbb{R}^D)$, as a function, cannot take infinite value. So the infinite supremum of the objective in (4) cannot be obtained by any element in $L^2(p; \mathbb{R}^D)$, thus no optimal solution for the optimization problem. □

A3: Derivation of the vector field of GFSF

The vector field $u(x) = -\nabla \log q(x)$ in GFSF is identified by the optimization problem (2):

$$\min_{u} \max_{\phi \in \mathcal{H}^D, \|\phi\|=1} \Big(\sum_j \big(u^{(j)} \cdot \phi(x^{(j)}) - \nabla \cdot \phi(x^{(j)}) \big) \Big)^2,$$

where $u^{(j)} := u(x^{(j)})$. For ϕ in $\mathcal{H}^D$, by using the reproducing property $\langle \phi_\alpha(\cdot), K(x, \cdot) \rangle_{\mathcal{H}} = \phi_\alpha(x)$ and $\langle \phi_\alpha(\cdot), \partial_{x_\beta} K(x, \cdot) \rangle_{\mathcal{H}} = \partial_{x_\beta} \phi_\alpha(x)$ [38], we can write the objective function as

$$\begin{aligned}
&\Big(\sum_\alpha \sum_j \big(u_\alpha^{(j)} \phi_\alpha(x^{(j)}) - \partial_{x_\alpha^{(j)}} \phi_\alpha(x^{(j)}) \big) \Big)^2 \\
&= \left(\sum_\alpha \left\langle \sum_j \big(u_\alpha^{(j)} K(x^{(j)}, \cdot) - \partial_{x_\alpha^{(j)}} K(x^{(j)}, \cdot) \big), \phi_\alpha(\cdot) \right\rangle_{\mathcal{H}} \right)^2 \\
&= \left\langle \sum_j \big(u^{(j)} K(x^{(j)}, \cdot) - \nabla_{x^{(j)}} K(x^{(j)}, \cdot) \big), \phi(\cdot) \right\rangle_{\mathcal{H}^D}^2 .
\end{aligned}$$

We denote $\zeta := \sum_j \big(u^{(j)} K(x^{(j)}, \cdot) - \nabla_{x^{(j)}} K(x^{(j)}, \cdot) \big) \in \mathcal{H}^D$. Then the optimal value of the objective after maximizing out ϕ is $\|\zeta\|_{\mathcal{H}^D}^2 = \sum_{i,j} \big(u^{(i)} u^{(j)} K(x^{(i)}, x^{(j)}) - 2u^{(i)} \nabla_{x^{(j)}} K(x^{(j)}, x^{(i)}) + \nabla_{x^{(i)}} \nabla_{x^{(j)}} K(x^{(i)}, x^{(j)}) \big) = \mathrm{tr}(\hat{u} \hat{K} \hat{u}^\top) - 2\mathrm{tr}(\hat{K}' \hat{u}^\top) + \mathrm{const}$, which should be minimized wrt u. By further differentiating wrt each component of $\hat{u}$, the optimal solution of $\hat{u}$ should be $\hat{u} = \hat{K}' \hat{K}^{-1}$.

A4: Scalar-valued test function $\varphi \in \mathcal{C}_c^\infty(\mathcal{X})$ for GFSF

For the equality $u(x) = -\nabla \log q(x)$, or $u(x)q(x) + \nabla q(x) = 0$, to hold in the distributional sense with scalar-valued test function, we mean

$$\int_{\mathbb{R}^D} \big(\varphi(x) u(x) - \nabla \varphi(x) \big) q(x) \mathrm{d}x = 0, \forall \varphi \in \mathcal{C}_c^\infty(\mathcal{X}). \tag{12}$$

Let $\{x^{(j)}\}_j$ be a set of samples of $q(x)$. Then the above requirement on $u(x)$ is

$$\sum_j \Big(\varphi(x^{(j)}) u^{(j)} - \nabla \varphi(x^{(j)}) \Big) = 0, \forall \varphi \in \mathcal{C}_c^\infty(\mathcal{X}), \tag{13}$$

where $u^{(j)} = u(x^{(j)})$. As analyzed above, for a valid vector field, we have to smooth the function φ.

For the above considerations, we restrict φ in Eq. (13) to be in the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ of some kernel $K(\cdot, \cdot)$, and convert the equation as the following optimization problem:

$$\min_{u} \max_{\varphi \in \mathcal{H}, \|\varphi\|_{\mathcal{H}}=1} J(u, \varphi) := \sum_{j,\alpha} \Big(\varphi(x^{(j)}) u_\alpha^{(j)} - \partial_{x_\alpha^{(j)}} \varphi(x^{(j)}) \Big)^2 . \tag{14}$$

By using the reproducing properties of RKHS, we can write $J(u, \varphi)$ as

$$J(u, \varphi) = \sum_\alpha \langle \varphi(\cdot), \zeta_\alpha(\cdot) \rangle_{\mathcal{H}}^2, \quad \zeta_\alpha(\cdot) := \sum_j \Big(u_\alpha^{(j)} K(x^{(j)}, \cdot) - \partial_{x_\alpha^{(j)}} K(x^{(j)}, \cdot) \Big) .$$

By linear algebra operations, we have

$$\max_{\varphi \in \mathcal{H}, \|\varphi\|_{\mathcal{H}}=1} J(u, \varphi) = \lambda_1(A(u)),$$

where $\lambda_1(A(u))$ is the largest eigenvalue of matrix A, where $A(u)_{\alpha\beta} = \langle \zeta_\alpha(\cdot), \zeta_\beta(\cdot) \rangle_{\mathcal{H}}$, or

$$A(u) = \hat{u} \hat{K} \hat{u}^\top - (\hat{K}' \hat{u}^\top + \hat{u} \hat{K}'^\top) + \hat{K}'',$$

where $\hat{K}''_{\alpha\beta} := \sum_{i,j} \partial_{x_\alpha^{(i)}} \partial_{x_\beta^{(j)}} K(x^{(i)}, x^{(j)})$. For distinct samples $\hat{K}$ is positive-definite, so we can conduct Cholesky decomposition: $\hat{K} = GG^\top$ with G non-singular. Note that $A(u) = (\hat{u}G - \hat{K}' G^{-1\top})(\hat{u}G - \hat{K}' G^{-1\top})^\top + (\hat{K}'' - \hat{K}' \hat{K}^{-1} \hat{K}'^\top)$. So whenever $\hat{u}G \neq \hat{K}' G^{-1\top}$, the first term will be positive semidefinite with positive largest eigenvalue, which makes $\lambda_1(A(u)) > \lambda_1(\hat{K}'' - \hat{K}' \hat{K}^{-1} \hat{K}'^\top)$. So to minimize $\lambda_1(A(u))$, we require $\hat{u}G = \hat{K}' G^{-1\top}$, i.e. $\hat{u} = \hat{K}'(GG^\top)^{-1} = \hat{K}' \hat{K}^{-1}$, which coincides with the result for vector-valued test function $\phi \in \mathcal{H}^D$.

A5: Details on the HE method for bandwidth selection

We first note that the bandwidth selection problem cannot be solved using theories of heat kernels, which aims to find the evolving density under the Brownian motion with known initial distribution, while in our case the density is unknown and we want to find an update on samples to approximate the effect of Brownian motion.

For $\tilde{q}(x;\{x^{(j)}\}_j) = (1/Z)\sum_j c(\|x-x^{(j)}\|^2/(2h))$, the minimizing objective Eq. (3) becomes

$$\sum_k \Bigg(\sum_j \Bigg[c_j''(x)\|x-x^{(j)}\|^2 + Dhc_j'(x) + \frac{\big(\sum_i c_{ij}' x^{(i)}\big) - \big(\sum_i c_{ij}'\big)x^{(j)}}{\big(\sum_i c_{ij}\big)} \cdot (x-x^{(j)})c_j'(x) \Bigg] \Bigg)^2 = 0,$$

where $c_j'(x) = c'(\|x-x^{(j)}\|^2/(2h))$, $c_{ij}' = c_j'(x^{(i)})$, $c_{ij} = c(\|x^{(i)}-x^{(j)}\|^2/(2h))$. Then for Gaussian kernel $c(r) = (2\pi h)^{-\frac{D}{2}} e^{-r}$, denoting $g_k^2(h)$ as the summand for k of the l.h.s. of the above equation, we have

$$\begin{aligned}
(2\pi)^{\frac{D}{2}} g_k(h) =& \big(\sum_j e_{kj}\|d_{kj}\|^2\big) - hD\big(\sum_j e_{kj}\big) - \sum_j \big(\sum_i e_{ij}\big)^{-1} e_{jk} d_{jk} \cdot \big(\sum_i e_{ij} d_{ij}\big), \\
(2\pi)^{\frac{D}{2}} g_k'(h) =& \ \frac{1}{2h^2}\big(\sum_j e_{jk}\|d_{jk}\|^4\big) - \frac{D}{h}\big(\sum_j e_{jk}\|d_{jk}\|^2\big) + \big(\frac{D^2}{2} - D\big)\big(\sum_j e_{jk}\big) \\
& - \frac{1}{2h^2}\sum_j \big(\sum_i e_{ij}\big)^{-1} e_{jk} d_{jk} \cdot \big(\sum_i e_{ij}\|d_{ij}\|^2 d_{ij}\big) \\
& - \frac{1}{2h^2}\sum_j \big(\sum_i e_{ij}\big)^{-1} e_{jk}\|d_{jk}\|^2 d_{jk} \cdot \big(\sum_i e_{ij} d_{ij}\big) \\
& + \frac{1}{2h^2}\sum_j \big(\sum_i e_{ij}\big)^{-2} \big(\sum_i e_{ij}\|d_{ij}\|^2\big) e_{jk} d_{jk} \cdot \big(\sum_i e_{ij} d_{ij}\big) \\
& + \frac{D}{2h}\sum_j \big(\sum_i e_{ij}\big)^{-1} e_{jk} d_{jk} \cdot \big(\sum_i e_{ij} d_{ij}\big),
\end{aligned}$$

where $d_{ij} = x^{(i)} - x^{(j)}$, $e_{ij} = e^{-\|d_{ij}\|^2/(2h) - (D/2)\log h}$. Then we can choose h by minimizing $\sum_k g_k^2(h)$. Although the evaluation of $g_k(h)$ may induce some computation cost, the optimization is wrt a scalar, where we can use efficient line-search methods and only a few (e.g. five) iterations are needed in each particle updating step, with initialization using the value of the last step.

A6: Details on the Wasserstein Nesterov's accelerated gradient method

A6.1: Details on the parallel transport on $\mathcal{P}_2$

We follow the Schild's ladder method to parallel transport a tangent vector at $\boldsymbol{q_1}$, $\xi \in T_{\boldsymbol{q_1}}\mathcal{P}_2$, to the tangent space at $\boldsymbol{q_2}$, $T_{\boldsymbol{q_2}}\mathcal{P}_2$. Assume $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$ are close in the sense of the 2-Wasserstein distance, so that the Schild's ladder finds a good first-order approximation. In the following we consider transporting $\varepsilon\xi$ for small $\varepsilon > 0$ for the sake of the pairwisely close condition, and the result can be recovered by noting the linearity of the parallel transport: $\Gamma_{\boldsymbol{q_1}}^{\boldsymbol{q_2}}(\varepsilon\xi) = \varepsilon\Gamma_{\boldsymbol{q_1}}^{\boldsymbol{q_2}}(\xi)$. We adopt the vector field form of the tangent vector ξ. Let $\{x_1^{(j)}\}_{j=1}^N$ and $\{x_2^{(j)}\}_{j=1}^N$ be the sets of samples of $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$, respectively, and assume that they are pairwisely close.

The measure $\mathrm{Exp}_{\boldsymbol{q_1}}(\varepsilon\xi)$ can be identified as $(\mathrm{id} + \varepsilon\xi)_{\#}\boldsymbol{q_1}$ due to the knowledge on the exponential map on $\mathcal{P}_2$ explained in Section 4, thus $\{x_1^{(j)} + \varepsilon\xi(x_1^{(j)})\}_{j=1}^N$ is a set of samples of $\mathrm{Exp}_{\boldsymbol{q_1}}(\varepsilon\xi)$ (see Lemma 2), and still pairwisely close to $\{x_2^{(j)}\}_j$ for small enough ε. So we know that the optimal map $\mathcal{T}$ from $\boldsymbol{q_2}$ to $\mathrm{Exp}_{\boldsymbol{q_1}}(\varepsilon\xi)$ satisfies $\mathcal{T}(x_2^{(j)}) = x_1^{(j)} + \varepsilon\xi(x_1^{(j)})$, and according to Theorem 7.2.2 of [1], the geodesic from $\boldsymbol{q_2}$ to $\mathrm{Exp}_{\boldsymbol{q_1}}(\varepsilon\xi)$ is $t \mapsto \big((1-t)\mathrm{id} + t\mathcal{T}\big)_{\#}\boldsymbol{q_2}$. Thus a set of samples of $\boldsymbol{q_M} = \frac{1}{2}(\mathrm{id} + \mathcal{T})_{\#}\boldsymbol{q_2}$, i.e. the midpoint of the geodesic, can be derived as $\Big\{\frac{1}{2}\big(x_2^{(j)} + x_1^{(j)} + \varepsilon\xi(x_1^{(j)})\big)\Big\}_j$. Following a similar procedure, a set of samples of $\boldsymbol{q_E}$ is found as

$\left\{(1-t)x_1^{(j)} + \frac{1}{2}t\big(x_2^{(j)} + x_1^{(j)} + \varepsilon\xi(x_1^{(j)})\big)\right\}_j\Big|_{t=2} = \left\{x_2^{(j)} + \varepsilon\xi(x_1^{(j)})\right\}_j$ and is pairwisely close to $\{x_2^{(j)}\}_j$. Thus the approximated transported tangent vector $\tilde{\Gamma}_{\boldsymbol{q_1}}^{\boldsymbol{q_2}}(\varepsilon\xi) = \mathrm{Exp}_{\boldsymbol{q_2}}^{-1}(\boldsymbol{q}_E)$ satisfies $\big(\tilde{\Gamma}_{\boldsymbol{q_1}}^{\boldsymbol{q_2}}(\varepsilon\xi)\big)(x_2^{(j)}) = \varepsilon\xi(x_1^{(j)})$, thus $\big(\tilde{\Gamma}_{\boldsymbol{q_1}}^{\boldsymbol{q_2}}(\xi)\big)(x_2^{(j)}) = \xi(x_1^{(j)})$.

A6.2: Details on deriving the Wasserstein Nesterov's accelerated gradient method (WNAG)

We adopt Alg. 2 of [20], which accommodates for more general objective functions. Before iteration k, let $\boldsymbol{q}_{k-1}$ be the current measure of interest with samples $\{x_{k-1}^{(j)}\}_{j=1}^N$, and $\boldsymbol{r}_{k-1}$ be the current auxiliary measure with samples $\{y_{k-1}^{(j)}\}_{j=1}^N$. Assume the two measures are close and the two sets of samples are pairwisely close, which naturally hold true for $k=1$ due to our initialization $y_0^{(j)} = x_0^{(j)}, \forall j$.

According to Alg. 2 of [20], $\boldsymbol{q}$ is updated by $\boldsymbol{q}_k = \mathrm{Exp}_{\boldsymbol{r}_{k-1}}(\varepsilon\xi_{k-1})$, where ξ_{k-1} is the (Riemannian) gradient (in vector field form) of the objective function on $\mathcal{P}_2$ at $\boldsymbol{r}_{k-1}$, and can be estimated by various methods like SVGD, GFSD and GFSF. As stated in Section 4, $\mathrm{Exp}_{\boldsymbol{r}_{k-1}}(\varepsilon\xi_{k-1}) = (\mathrm{id} + \varepsilon\xi_{k-1})_{\#}\boldsymbol{r}_{k-1}$, which indicates that $\{y_{k-1}^{(j)} + \varepsilon\xi_{k-1}^{(j)}\}_{j=1}^N$ where $\xi_{k-1}^{(j)} := \xi_{k-1}(y_{k-1}^{(j)})$ is a set of samples of the mapped measure (see Lemma 2). Thus we assign $x_k^{(j)} = y_{k-1}^{(j)} + \varepsilon\xi_{k-1}^{(j)}$.

The update of $\boldsymbol{r}$ is a little sophisticated. With the acceleration factor $\alpha > 3$ and an upper bound $D > 0$ on the diameter of $\mathcal{P}_2$, $\boldsymbol{r}_k$ is determined by the following equality that holds in $T_{\boldsymbol{r}_{k-1}}\mathcal{P}_2$ (see Eq. (5) of [20]):

$$\Gamma_{\boldsymbol{r}_k}^{\boldsymbol{r}_{k-1}}\left(\frac{k}{\alpha-1}\mathrm{Exp}_{\boldsymbol{r}_k}^{-1}(\boldsymbol{q}_k) + \frac{D\xi_k}{\|\xi_k\|_{\boldsymbol{r}_k}}\right) = \frac{k-1}{\alpha-1}\mathrm{Exp}_{\boldsymbol{r}_{k-1}}^{-1}(\boldsymbol{q}_{k-1}) + \frac{D\xi_{k-1}}{\|\xi_{k-1}\|_{\boldsymbol{r}_{k-1}}} - \frac{k+\alpha-2}{\alpha-1}\varepsilon\xi_{k-1}.$$

To simplify the equality, we replace $\Gamma_{\boldsymbol{r}_k}^{\boldsymbol{r}_{k-1}}(\xi_k)$ by ξ_{k-1}, which is also done in the original paper [20]. Also note that $\|\xi_k\|_{\boldsymbol{r}_k} = \|\Gamma_{\boldsymbol{r}_k}^{\boldsymbol{r}_{k-1}}(\xi_k)\|_{\boldsymbol{r}_{k-1}}$, and $\big(\Gamma_{\boldsymbol{r}_k}^{\boldsymbol{r}_{k-1}}\big)^{-1} = \Gamma_{\boldsymbol{r}_{k-1}}^{\boldsymbol{r}_k}$. Then the equality can be simplified as

$$\frac{k}{\alpha-1}\mathrm{Exp}_{\boldsymbol{r}_k}^{-1}(\boldsymbol{q}_k) = \Gamma_{\boldsymbol{r}_{k-1}}^{\boldsymbol{r}_k}\left(\frac{k-1}{\alpha-1}\mathrm{Exp}_{\boldsymbol{r}_{k-1}}^{-1}(\boldsymbol{q}_{k-1}) - \frac{k+\alpha-2}{\alpha-1}\varepsilon\xi_{k-1}\right).$$

By further noting that $\mathrm{Exp}_{\boldsymbol{r}_k}^{-1}(\boldsymbol{q}_k) = -\Gamma_{\boldsymbol{q}_k}^{\boldsymbol{r}_k}(\mathrm{Exp}_{\boldsymbol{q}_k}^{-1}(\boldsymbol{r}_k))$ and approximating $\Gamma_{\boldsymbol{r}_k}^{\boldsymbol{q}_k}\Gamma_{\boldsymbol{r}_{k-1}}^{\boldsymbol{r}_k}$ by $\Gamma_{\boldsymbol{r}_{k-1}}^{\boldsymbol{q}_k}$, we have

$$\boldsymbol{r}_k = \mathrm{Exp}_{q_k}\left(-\Gamma_{\boldsymbol{r}_{k-1}}^{\boldsymbol{q}_k}(\zeta_{k-1})\right), \quad \zeta_{k-1} = \frac{k-1}{k}\mathrm{Exp}_{\boldsymbol{r}_{k-1}}^{-1}(\boldsymbol{q}_{k-1}) - \frac{k+\alpha-2}{k}\varepsilon\xi_{k-1}.$$

By assumption, $\{x_{k-1}^{(j)}\}_{j=1}^N$ of $\boldsymbol{q}_{k-1}$ and $\{y_{k-1}^{(j)}\}_{j=1}^N$ of $\boldsymbol{r}_{k-1}$ are pairwisely close, so from Section 4 we know that $\mathrm{Exp}_{\boldsymbol{r}_{k-1}}^{-1}(\boldsymbol{q}_{k-1})(y_{k-1}^{(j)}) = x_{k-1}^{(j)} - y_{k-1}^{(j)}$, thus $\zeta_{k-1}(y_{k-1}^{(j)}) = \frac{k-1}{k}(x_{k-1}^{(j)} - y_{k-1}^{(j)}) - \frac{k+\alpha-2}{k}\varepsilon\xi_{k-1}^{(j)}$. Due to the update rule for $\boldsymbol{q}_k$ that we already discovered: $x_k^{(j)} = y_{k-1}^{(j)} + \varepsilon\xi_{k-1}^{(j)}$, we know that $\{x_k^{(j)}\}_{j=1}^N$ of $\boldsymbol{q}_k$ and $\{y_{k-1}^{(j)}\}_{j=1}^N$ of $\boldsymbol{r}_{k-1}$ are pairwisely close, for small enough step size ε. So from Section 4, we know that the approximation to $\Gamma_{\boldsymbol{r}_{k-1}}^{\boldsymbol{q}_k}(\zeta_{k-1})$ by the Schild's ladder method, satisfies $\big(\tilde{\Gamma}_{\boldsymbol{r}_{k-1}}^{\boldsymbol{q}_k}(\zeta_{k-1})\big)(x_k^{(j)}) = \zeta_{k-1}(y_{k-1}^{(j)})$. Finally, we assign $y_k^{(j)} = x_k^{(j)} - \big(\tilde{\Gamma}_{\boldsymbol{r}_{k-1}}^{\boldsymbol{q}_k}(\zeta_{k-1})\big)(x_k^{(j)}) = x_k^{(j)} - \zeta_{k-1}(y_{k-1}^{(j)}) = x_k^{(j)} - \frac{k-1}{k}(x_{k-1}^{(j)} - y_{k-1}^{(j)}) + \frac{k+\alpha-2}{k}\varepsilon\xi_{k-1}^{(j)}$ as a set of samples of $\boldsymbol{r}_k$. We note that by assumption $\{x_{k-1}^{(j)}\}_{j=1}^N$ and $\{y_{k-1}^{(j)}\}_{j=1}^N$ are pairwisely close, so for sufficiently small ε, $\zeta_{k-1}(y_{k-1}^{(j)})$ is an infinitesimal vector for all j. This, in turn, indicates that $\{x_k^{(j)}\}_{j=1}^N$ of $\boldsymbol{q}_k$ and $\{y_k^{(j)}\}_{j=1}^N$ of $\boldsymbol{r}_k$ are pairwisely close, which provides the assumption for the next iteration. Now the derivation of our WNAG method, i.e. our Alg. 1, is completed.

A7: More results on the experiment on the Bayesian neural network

The results on other datasets that [18] uses on the Bayesian neural network experiment are shown in Table 2. Still the settings are identical to the experiments of Liu *et al.* [18], except we run all methods for a same amount of iterations on every dataset. We find that the WNAG method achieves a salient improvement over the WGD method and outperforms the PO methods, and GFSD/GFSF methods achieve better results than SVGD in most cases.

Table 2: Results on Bayesian neural network on other datasets. Following the settings of [18], we average the results over 20 runs with random 90%train-10%test split of every dataset and the mini-batch size is taken as 100.

(*Dataset*) Method	Avg. Test RMSE			Avg. Test LL		
	SVGD	GFSD (Ours)	GFSF (Ours)	SVGD	GFSD (Ours)	GFSF (Ours)
(*Concrete*) WGD	5.324±0.104	5.234±0.114	5.060±0.131	-3.082±0.018	-3.042±0.016	-3.046±0.026
PO	5.138±0.251	4.819±0.243	4.783±0.264	-3.063±0.036	-3.004±0.024	-2.990±0.037
WNAG (Ours)	4.664±0.197	**4.238±0.316**	4.699±0.272	**-2.824±0.023**	-2.893±0.033	-2.917±0.033
(*Energy*) WGD	1.374±0.045	1.079±0.109	1.052±0.105	-1.767±0.024	-1.537±0.048	-1.514±0.099
PO	0.566±0.060	0.535±0.062	0.527±0.069	-0.869±0.099	-0.829±0.070	-0.821±0.097
WNAG (Ours)	**0.375±0.041**	0.378±0.039	0.388±0.053	**-0.540±0.058**	-0.573±0.067	-0.557±0.068
(*Naval*) WGD	(4.2±0.2)e-3	(3.9±0.1)e-3	(4.0±0.1)e-3	4.089±0.012	4.105±0.027	4.089±0.026
PO	(3.6±0.3)e-3	(2.6±0.1)e-3	(1.9±0.1)e-3	4.181±0.048	4.783±0.028	4.875±0.039
WNAG (Ours)	(2.4±0.3)e-3	(1.3±0.1)e-3	**(0.9±0.1)e-3**	4.714±0.036	5.080±0.017	**5.159±0.030**
(*Combined*) WGD	4.033±0.033	3.974±0.035	3.995±0.026	-2.819±0.008	-2.810±0.012	-2.810±0.009
PO	4.090±0.036	4.058±0.031	3.894±0.030	-2.837±0.019	-2.820±0.014	-2.790±0.013
WNAG (Ours)	3.903±0.022	3.905±0.025	**3.890±0.028**	-2.786±0.012	**-2.767±0.009**	-2.794±0.011
(*Wine*) WGD	0.609±0.010	0.601±0.012	0.612±0.013	-0.925±0.014	-0.929±0.018	-0.941±0.012
PO	0.594±0.014	0.602±0.021	0.608±0.014	-0.903±0.021	-0.935±0.020	-0.933±0.021
WNAG (Ours)	**0.559±0.023**	0.568±0.020	0.576±0.013	**-0.849±0.026**	-0.865±0.023	-0.876±0.017