Statistics and Computing (2021) 31:34
https://doi.org/10.1007/s11222-021-10009-7

Particle-based energetic variational inference

Yiwei Wang[1] · Jiuhai Chen[1] · Chun Liu[1] · Lulu Kang[1]

Received: 2 June 2020 / Accepted: 23 March 2021

Abstract
We introduce a new variational inference (VI) framework, called *energetic variational inference* (EVI). It minimizes the VI objective function based on a prescribed *energy-dissipation law*. Using the EVI framework, we can derive many existing particle-based variational inference (ParVI) methods, including the popular Stein variational gradient descent (SVGD). More importantly, many new ParVI schemes can be created under this framework. For illustration, we propose a new particle-based EVI scheme, which performs the particle-based approximation of the density first and then uses the approximated density in the variational procedure, or "Approximation-then-Variation" for short. Thanks to this order of approximation and variation, the new scheme can maintain the variational structure at the particle level, and can significantly decrease the KL-divergence in each iteration. Numerical experiments show the proposed method outperforms some existing ParVI methods in terms of fidelity to the target distribution.

Keywords KL-divergence · Energetic variational approach · Gaussian mixture model · Kernel function · Implicit-Euler · Variational inference

1 Introduction

Bayesian methods play an important role in statistics and data science nowadays. They provide a rigorous framework for uncertainty quantification of various statistical learning models (Stuart 2010; Gelman et al. 2013). The main components of a Bayesian model include a set of observational data $\{\boldsymbol{y}_i\}_{i=1}^I$ with $\boldsymbol{y}_i \in \mathbb{R}^D$, the model assumption of the likelihood $\rho(\{\boldsymbol{y}_i\}_{i=1}^I|\boldsymbol{x})$ with certain unknown parameters $\boldsymbol{x} \in \mathbb{R}^d$, and a user-specified prior distribution for the parameters $\rho_0(\boldsymbol{x})$. The key step in Bayesian inference is to obtain the posterior distribution, denoted by $\rho(\boldsymbol{x}|\{\boldsymbol{y}_i\}_{i=1}^I)$. Following the Bayes' theorem, the posterior distribution of the unknown parameters is

$$\rho(\boldsymbol{x}|\{\boldsymbol{y}_i\}_{i=1}^I) = \frac{\rho(\{\boldsymbol{y}_i\}_{i=1}^I|\boldsymbol{x})\rho_0(\boldsymbol{x})}{\rho(\{\boldsymbol{y}_i\}_{i=1}^I)}.$$

However, it is a long standing challenge to obtain the posterior distribution in practice when the analytical formula of $\rho(\boldsymbol{x}|\{\boldsymbol{y}_i\}_{i=1}^I)$ is not tractable due to the integration $\rho(\{\boldsymbol{y}_i\}_{i=1}^I) = \int \rho(\{\boldsymbol{y}_i\}_{i=1}^I|\boldsymbol{x})\rho_0(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$.

Many approximate inference methods have been developed to approximate the posterior distribution. Among them, two popular classes of methods are Markov Chain Monte Carlo (MCMC) algorithms (Metropolis et al. 1953; Hastings 1970; Geman and Geman 1984; Welling and Teh 2011) and Variational Inference (VI) methods (Jordan et al. 1999; Neal and Hinton 1998; Wainwright and Jordan 2008; Blei et al. 2017). MCMC is a family of methods that generate samples by constructing a Markov chain whose equilibrium distribution is the target distribution. Examples include the Metropolis–Hastings algorithm (Metropolis et al. 1953; Hastings 1970), Gibbs sampling (Geman and Geman 1984; Casella and George 1992), Langevin Monte Carlo (LMC) (Rossky et al. 1978; Parisi 1981; Roberts and Tweedie 1996; Welling and Teh 2011), and Hamiltonian Monte Carlo (HMC) (Neal 1993; Duane et al. 1987).

The VI framework essentially transforms the inference problem into an optimization problem, which minimizes some kind of objective functional over a prescribed family

✉ Lulu Kang
lkang2@iit.edu

Yiwei Wang
ywang487@iit.edu

Jiuhai Chen
jchen168@hawk.iit.edu

Chun Liu
cliu124@iit.edu

[1] Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL, USA

Published online: 17 April 2021

of distributions denoted by $\mathcal{Q}$ (Blei et al. 2017). The objective functional measures the difference between a candidate distribution in $\mathcal{Q}$ and the target distribution. For Bayesian models, the target distribution is the posterior distribution. VI has a wide application that goes beyond Bayesian statistics and is a powerful tool for approximating probability densities. A common choice of the objective functional is the Kullback–Leibler (KL) divergence (Blei et al. 2017). For any two distributions $\rho(\boldsymbol{x})$ and $\rho^*(\boldsymbol{x})$, the KL-divergence from ρ to ρ^* is given by

$$\mathrm{KL}(\rho(\boldsymbol{x})||\rho^*(\boldsymbol{x})) = \int \rho(\boldsymbol{x}) \ln\left(\frac{\rho(\boldsymbol{x})}{\rho^*(\boldsymbol{x})}\right) \mathrm{d}\boldsymbol{x}. \tag{1.1}$$

In the review paper (Blei et al. 2017), the authors have made a detailed comparison between MCMC and VI methods, both conceptually and numerically. Essentially, MCMC methods guarantee the convergence of the generated samples to the target distribution when certain conditions are met. But the price of this asymptotic property is that MCMC methods tend to be more computationally intensive and thus might not be suitable for large datasets. On the contrary, VI methods do not have this asymptotic guarantee. For particle-based VI, the fidelity of the empirical distribution of the particles to the target distribution depends on the VI algorithm as well as the number of particles. On the other hand, since VI is essentially an optimization problem and it can take advantage of the stochastic optimization methods, VI methods can be significantly faster than MCMC. For detailed differences and connections between the two types of methods, see MacKay and Mac Kay (2003) and Salimans et al. (2015). In this paper, we only focus on the VI approaches.

The VI framework minimizes $\mathrm{KL}(\rho(\boldsymbol{x})||\rho^*(\boldsymbol{x}))$ with respect to $\rho \in \mathcal{Q}$ in order to approximate the target distribution ρ^*. In traditional VI methods (Blei et al. 2017), $\mathcal{Q}$ is often taken as a family of parametric distributions. There also have been growing interests in flow-based VI methods, in which $\mathcal{Q}$ consists of distributions obtained by a series of smooth transformations from a tractable initial reference distribution. Examples include normalizing flow VI methods (Rezende and Mohamed 2015; Kingma et al. 2016; Salman et al. 2018) and particle-based VI methods (ParVIs) (Liu and Wang 2016; Liu 2017; Liu and Zhu 2018; Chen et al. 2018; Liu et al. 2019; Chen et al. 2019). One ParVI method that has attracted much attention is the Stein Variational Gradient Descent (SVGD) (Liu and Wang 2016; Detommaso et al. 2018; Wang et al. 2019; Li et al. 2019). Many existing ParVI methods can be viewed as some versions of the approximated Wasserstein gradient flow of the KL-divergence (Liu 2017). As explained in Sect. 3, these methods may not preserve the variational structure at the particle level because approximation of the density function is performed after the variational step.

In this paper, we introduce a new variational inference framework, named as *Energetic Variational Inference* (EVI). It consists of two ingredients, a continuous formulation of the variational inference, and a discretization strategy that leads to a practical algorithm. Inspired by the non-equilibrium thermodynamics, we propose using an *energy-dissipation law* to describe the mechanism of minimizing the VI objective functional, for instance, the KL-divergence. An energetic variational inference algorithm can be obtained by employing an energetic variational approach and a proper discretization. Using the EVI framework, we can derive and explain many existing ParVI methods, such as the SVGD method. More importantly, many new ParVI schemes can be created under the EVI framework. To demonstrate how to develop a new EVI method, we propose a new particle-based EVI scheme, which performs the particle-based approximation of the density first and then uses the approximated density in the variational procedure. Thanks to this "Approximation-then-Variation" order, we can derive a system of ordinary differential equations (ODEs) of particles that preserves the variational structure at the particle level, which is different from many existing methods. Such an ODE system can be solved via the implicit Euler method, which can be reformulated into an optimization problem. By the virtue of the variational structure at the particle level, we can significantly decrease the discretized KL-divergence in every iteration and push the density of the particles close to the target distribution efficiently.

In the remaining sections, we first introduce some preliminary background on the flow map and the energetic variational approach that is commonly used in mathematical modeling in Sect. 2. In Sect. 3, we propose the energetic variational inference (EVI) framework. Specifically, we first lay out the general continuous formulation of the EVI, and then introduce two different ways to discretize the continuous EVI. One is the "Approximation-then-Variation" approach and the other is the "Variation-then-Approximation" approach. Both lead to particle-based EVI methods. The dynamics of the particles are described by an ODE system, which can be solved by explicit or implicit Euler methods. Using the implicit-Euler, we propose one new example of particle-based EVI, called EVI-Im. In Sect. 4, we compare the EVI-Im with some existing particle-based VI methods. The paper is concluded in Sect. 5.

2 Preliminary

Before reviewing the preliminary topics on flow maps and the energetic variational approach, we first clarify some notations used in this paper. Let $f(\boldsymbol{x}, t)$ be a scalar function of d-dimensional space variable $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ and time $t \in [0, \infty)$. We denote the derivative of $f(\boldsymbol{x}, t)$ with respect

to t as $\dot{f}(\boldsymbol{x}, t)$ or $\dot{f}$ for short, and thus $\dot{f}$ is still a scalar function. The gradient of $f(\boldsymbol{x}, t)$ with respect to $\boldsymbol{x}$ is $\nabla f(\boldsymbol{x}, t)$ or ∇f, and thus ∇f is a d-dimensional function. When the time t is taken at a series of discrete values, i.e., $t = 0, 1, 2, \ldots$, the discrete t becomes the index of a sequence of function values of f evaluated at $\boldsymbol{x}$. We write the integer index in the superscript position of f, i.e., $f^t(\boldsymbol{x})$. The subscript position of f is to label different functions.

2.1 Minimizing KL-divergence through flow maps

The goal of the variational inference is to find a density function ρ from a family of density functions $\mathcal{Q}$ by minimizing the VI objective functional, such as the KL-divergence from $\rho(\boldsymbol{x})$ to the target density function $\rho^*(\boldsymbol{x})$. The complexity of this optimization problem is decided by the feasible region, i.e., the family $\mathcal{Q}$. Traditional variational inference methods choose $\mathcal{Q}$ as a parametric family of probability distributions. For example, the mean-field variational family assumes the mutual independence between the d dimensions of random variable $\boldsymbol{x}$, i.e., $\rho(\boldsymbol{x}) = \prod_{i=1}^{d} \rho_i(x_i)$, where ρ_i is a density function from a user specified family of one-dimensional probability densities (Bishop 2006; Blei et al. 2017).

In the flow-based VI methods, the set $\mathcal{Q}$ consists of distributions obtained by smooth transformations of a tractable initial reference distribution (Li et al. 2019). The idea of using maps to transform a distribution to another has been explored in many earlier papers (Tabak and Vanden-Eijnden 2010; El Moselhy and Marzouk 2012). Specifically, given a tractable reference distribution $\rho_0(z) : \mathcal{X}^0 \to \mathbb{R}^+$ and a sufficiently smooth one-to-one map $\boldsymbol{\phi}(\cdot)$, such that $\boldsymbol{x} = \boldsymbol{\phi}(z)$, the family $\mathcal{Q}$ is defined by

$$\mathcal{Q} = \{\rho_{[\boldsymbol{\phi}]}(\boldsymbol{x}) = \rho_0(\boldsymbol{\phi}^{-1}(\boldsymbol{x}))\left|\det[\nabla_{\boldsymbol{x}}\boldsymbol{\phi}^{-1}(\boldsymbol{x})]\right|, \boldsymbol{x} = \boldsymbol{\phi}(z), \\ \boldsymbol{\phi} : \mathcal{X}^0 \to \mathcal{X} \text{ is a smooth one-to-one map.}\} \tag{2.1}$$

We assume $\mathcal{X}^0 = \mathcal{X} = \mathbb{R}^d$ throughout this paper, but all the results can be generalized to the case where $\mathcal{X}^0 \neq \mathcal{X}$. Moreover, since $\boldsymbol{\phi}$ is one-to-one, we can enforce $\det[\nabla_z \boldsymbol{\phi}(z)] > 0$.

Given $\mathcal{Q}$ in (2.1), solving the following problem

$$\rho_{\text{opt}} = \arg\min_{\rho \in \mathcal{Q}} \text{KL}(\rho||\rho^*) \tag{2.2}$$

is equivalent to finding the optimal smooth one-to-one map $\boldsymbol{\phi}_{\text{opt}}$ such that

$$\rho_{\text{opt}}(\boldsymbol{x}) = \rho_0(\boldsymbol{\phi}_{\text{opt}}^{-1}(\boldsymbol{x}))\det[\nabla_{\boldsymbol{x}}\boldsymbol{\phi}_{\text{opt}}^{-1}(\boldsymbol{x})].$$

As in many optimization approaches, we expect it requires a number of transformations, say K steps, to find the optimal map, or equivalently,

$$\boldsymbol{\phi}_{\text{opt}}(\cdot) = \boldsymbol{\psi}^K \circ \boldsymbol{\psi}^{K-1} \ldots \circ \boldsymbol{\psi}^1(\cdot).$$

Each $\boldsymbol{\psi}^t(\cdot)$ is a smooth and one-to-one map such that $\boldsymbol{x}^t = \boldsymbol{\psi}^t(\boldsymbol{x}^{t-1})$. At the tth step, suppose $\boldsymbol{\phi}^t(\cdot) = \boldsymbol{\psi}^t \circ \boldsymbol{\psi}^{t-1} \ldots \circ \boldsymbol{\psi}^1(\cdot)$ is a proper transform, then

$$\begin{aligned}\rho^t(\boldsymbol{x}^t) &= \rho^{t-1}((\boldsymbol{\psi}^t)^{-1}(\boldsymbol{x}^t))\det[\nabla(\boldsymbol{\psi}^t)^{-1}(\boldsymbol{x}^t)] \\ &= \rho_0((\boldsymbol{\phi}^t)^{-1}(\boldsymbol{x}^t))\det[\nabla(\boldsymbol{\phi}^t)^{-1}(\boldsymbol{x}^t)].\end{aligned}$$

Intuitively, the series of transformations should move the initial density ρ_0 closer and closer to the target density ρ^* and eventually achieve convergence in terms of the KL-divergence. Therefore, $\text{KL}(\rho^t||\rho^*)$ should be decreased after each step, i.e.,

$$\text{KL}(\rho^t||\rho^*) - \text{KL}(\rho^{t-1}||\rho^*) \leq 0.$$

If we generalize the meaning of t from the discrete step index to the continuous time $t \in [0, \infty)$, we can consider $\rho^t(\boldsymbol{x})$ as a density function evolving continuously with respect to time t. To emphasize this point, we use the notation $\rho(\boldsymbol{x}, t)$ instead of $\rho^t(\boldsymbol{x})$. Therefore, $\text{KL}(\rho(\boldsymbol{x}, t)||\rho^*(\boldsymbol{x}))$ should be decreased with respect to t, i.e.,

$$\frac{\text{d}}{\text{d}t}\text{KL}(\rho(\boldsymbol{x}, t)||\rho^*(\boldsymbol{x})) \leq 0.$$

The key to minimizing the KL-divergence is to determine the speed of decreasing $\text{KL}(\rho(\boldsymbol{x}, t)||\rho^*(\boldsymbol{x}))$. In Sect. 3, we show how to use an energy-dissipating law to specify $\frac{\text{d}}{\text{d}t}\text{KL}(\rho(\boldsymbol{x}, t)||\rho^*(\boldsymbol{x}))$.

When t is generalized to continuous time, $\boldsymbol{\phi}^t(\cdot)$ becomes a smooth one-to-one map that also continuously evolves. Therefore, we use the notation $\boldsymbol{\phi}(\cdot, t)$ instead of $\boldsymbol{\phi}^t(\cdot)$. Since $\boldsymbol{\phi}(\cdot, t)$ is a smooth one-to-one map, it can be defined through a smooth, bounded velocity field $\mathbf{u} \in \mathbb{R}^d \times [0, \infty)$ as in Definition 1. This definition is also used in Sonoda and Murata (2019).

Definition 1 Given a smooth and bounded velocity field $\mathbf{u} : \mathbb{R}^d \times [0, \infty) \to \mathbb{R}^d$, a flow map $\boldsymbol{\phi}(z, t) : \mathbb{R}^d \times [0, \infty) \to \mathbb{R}^d$ is a map specified by an ordinary differential equation (for any fixed z)

$$\begin{cases}\dot{\boldsymbol{\phi}}(z, t) = \mathbf{u}(\boldsymbol{\phi}(z, t), t), & z \in \mathbb{R}^d, \quad t > 0 \\ \boldsymbol{\phi}(z, 0) = z, & z \in \mathbb{R}^d.\end{cases} \tag{2.3}$$

In continuum mechanics, $\boldsymbol{\phi}(z, t)$ is known as the *flow map* (Temam and Miranville 2005; Gonzalez and Stuart 2008). To better illustrate the idea of the flow map, we plot it conceptually in Fig. 1. For fixed z, $\boldsymbol{\phi}(z, t)$ is the trajectory of a particle (or sample) with initial position z. For fixed t, $\boldsymbol{\phi}(z, t)$ is a diffeomorphism between $\mathcal{X}^0$ (the initial domain) and $\mathcal{X}^t$ (the domain after t transformations). An intuitive interpretation

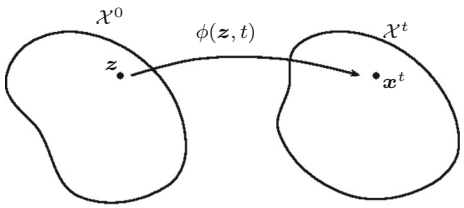

Fig. 1 An illustration of a flow map $\boldsymbol{\phi}(z, t)$

of $\mathbf{u}$ is that $\mathbf{u}$ is the speed of the probability mass, which is transported due to the transformation $\boldsymbol{\phi}(\cdot, t)$. But directly finding $\boldsymbol{\phi}(\cdot, t)$ is a difficult task. Thanks to this relationship (2.3), we can decide the transform $\boldsymbol{\phi}(\cdot, t)$ by specifying $\mathbf{u}$.

When the flow map is defined by the velocity field $\mathbf{u}(\boldsymbol{x}, t)$, the corresponding distribution $\rho(\boldsymbol{x}, t)$ is given by

$$\rho(\boldsymbol{\phi}(z, t), t) = \frac{\rho_0(z)}{\det[\nabla_z \boldsymbol{\phi}(z, t)]}.$$

The relation between $\mathbf{u}$ and $\rho(\boldsymbol{x}, t)$ is described by the following proposition.

Proposition 1 (Transportation equation) *If $\boldsymbol{\phi}(z, t)$ satisfies (2.3) and $\boldsymbol{x} = \boldsymbol{\phi}(z, t)$, the time-dependent probability density $\rho(\boldsymbol{x}, t)$, which is induced by $\boldsymbol{\phi}(z, t)$, satisfies the transport equation*

$$\begin{cases} \dot{\rho} + \nabla \cdot (\rho \mathbf{u}) = 0, \\ \rho(\boldsymbol{x}, 0) = \rho_0(\boldsymbol{x}), \end{cases} \tag{2.4}$$

where ρ_0 is the initial density and $\dot{\rho}$ is the derivative of $\rho(\boldsymbol{x}, t)$ with respect to t.

Equation (2.4) is known as the transport equation or continuity equation (Villani 2008; Sonoda and Murata 2019). Its derivation is in "Appendix B". Definition 1 and Proposition 1 indicate that we only need to determine the transport velocity $\mathbf{u}(\boldsymbol{x}, t)$ so as to determine the flow map $\boldsymbol{\phi}(z, t)$ and $\rho(\boldsymbol{\phi}(z, t), t)$.

2.2 Energetic variational approach

Here, we briefly introduce the energetic variational approach in mathematical modeling (Liu 2009; Giga et al. 2017), which is originated from the pioneering works of Rayleigh (1873), Onsager (1931a, b). It provides a unique way to determine the dynamics of a system via a prescribed *energy-dissipation law*

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}[\boldsymbol{\phi}] = -2\mathcal{D}[\boldsymbol{\phi}, \dot{\boldsymbol{\phi}}], \tag{2.5}$$

which describes how the total energy of the system decreases with respect to time. Here, $\mathcal{F}$ is the Helmholtz free energy, $-2\mathcal{D} \leq 0$ is the rate of energy dissipation, $\boldsymbol{\phi}$ is the state variable of the system, and $\dot{\boldsymbol{\phi}}$ is the derivative of $\boldsymbol{\phi}$ with respect to t. For a given energy-dissipation law (2.5), the energetic variational approach derives the dynamics of the system (or how energy $\mathcal{F}$ dissipates over time) through two variational procedure, the least action principle (LAP) and the maximum dissipation principle (MDP), which leads to

$$\frac{\delta \mathcal{D}}{\delta \dot{\boldsymbol{\phi}}} = -\frac{\delta \mathcal{F}}{\delta \boldsymbol{\phi}}, \tag{2.6}$$

where $\frac{\delta \mathcal{F}}{\delta \boldsymbol{\phi}}$ denotes the Fréchet derivative of $\mathcal{F}$ with respect to $\boldsymbol{\phi}$, defined as $(\frac{\delta \mathcal{F}}{\delta \boldsymbol{\phi}}, \boldsymbol{\psi}) = \lim_{\epsilon \to 0} \frac{\mathcal{F}(\boldsymbol{\phi} + \epsilon \boldsymbol{\psi}) - \mathcal{F}(\boldsymbol{\phi})}{\epsilon}$, and $\frac{\delta \mathcal{D}}{\delta \dot{\boldsymbol{\phi}}}$ denotes the Fréchet derivative of $\mathcal{D}$ with respect to $\dot{\boldsymbol{\phi}}$. More details on the energetic variational approach and the derivation of (2.6) are shown in "Appendix A".

3 Energetic variational inference

3.1 Continuous formulation

In this subsection, we first propose a continuous formulation of EVI. The idea is to specify the dynamics of minimizing KL-divergence via an *energy-dissipation law*, and we can employ the energetic variational approach to obtain the equation of the flow map $\boldsymbol{\phi}(z, t)$. More specifically, as an analogy to physics, the KL-divergence is viewed as the *Helmholtz free energy* (Murphy 2012), i.e., $\mathcal{F}[\boldsymbol{\phi}] = \mathrm{KL}(\rho(\boldsymbol{x}, t)||\rho^*)$. The free energy $\mathcal{F}$ depends on $\boldsymbol{\phi}$ since $\boldsymbol{x} = \boldsymbol{\phi}(z, t)$ as a result of the flow map. We can impose an energy-dissipation law

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{KL}(\rho(\boldsymbol{x}, t)||\rho^*) = -\int \eta(\rho)||\dot{\boldsymbol{\phi}}||^2 \mathrm{d}\boldsymbol{x}, \tag{3.1}$$

where $\mathcal{D} = \frac{1}{2}\int \eta(\rho)||\dot{\boldsymbol{\phi}}||^2 \mathrm{d}\boldsymbol{x}$. Because the flow map $\boldsymbol{\phi}$ can be defined by the velocity field $\mathbf{u}$ (Definition 1), thus

$$\mathcal{D} = \frac{1}{2}\int \eta(\rho)||\dot{\boldsymbol{\phi}}||^2 \mathrm{d}\boldsymbol{x} = \frac{1}{2}\int \eta(\rho)||\mathbf{u}||^2 \mathrm{d}\boldsymbol{x} \geq 0.$$

The functional $\eta(\rho)$ is a user-specified functional of ρ satisfying $\eta(\rho) > 0$ if $\rho > 0$. We denote $||\boldsymbol{a}|| = \sqrt{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{a}}$ for $\boldsymbol{a} \in \mathbb{R}^d$ as the l_2 norm of a vector.

Since $\rho(\boldsymbol{x}, t) = \rho_0(\boldsymbol{\phi}^{-1}(\boldsymbol{x}, t))$ is determined by $\boldsymbol{\phi}(z, t)$ for a given $\rho_0(z)$, the KL-divergence can be viewed as a functional of $\boldsymbol{\phi}$. By taking variation of KL with respect to $\boldsymbol{\phi}$ (see "Appendix C" for the detailed derivation), we can obtain

$$-\frac{\delta \mathrm{KL}(\rho_{[\boldsymbol{\phi}]}||\rho^*)}{\delta \boldsymbol{\phi}} = -(\nabla \rho + \rho \nabla V), \tag{3.2}$$

where $\rho_{[\boldsymbol{\phi}]}(\boldsymbol{x}, t) = \rho_0(\boldsymbol{\phi}^{-1}(\boldsymbol{x}, t))$ and $V = -\ln \rho^*$.

Meanwhile, taking variational of $\mathcal{D}$ with respect to $\dot{\boldsymbol{\phi}}$ yields

$$\frac{\delta \mathcal{D}}{\delta \dot{\boldsymbol{\phi}}} = \eta(\rho)\dot{\boldsymbol{\phi}}.$$

Then according to (2.6), $\dot{\boldsymbol{\phi}}$, i.e., the transport velocity $\mathbf{u}$ satisfies

$$\eta(\rho)\dot{\boldsymbol{\phi}} = -(\nabla\rho + \rho\nabla V). \tag{3.3}$$

This equation gives us the specification of the transport velocity $\mathbf{u}$ based on the energy-dissipation law (3.1). Thanks to the transport equation (2.4), ρ can be obtained from the specified $\mathbf{u}$. Therefore, (3.3) can be used to find the ρ that minimizes the KL-divergence in the admissible set. Indeed, combining (3.3) with the transport equation (2.4), we have

$$\dot{\rho} = \nabla \cdot \left(\frac{\rho}{\eta(\rho)} (\nabla\rho + \rho\nabla V) \right), \tag{3.4}$$

which is the continuous differential equation formulation for ρ. One can choose $\eta(\rho)$ to control the dynamics of the system. In the remainder of the paper, we choose $\eta(\rho) = \rho$, which is consistent with Wasserstein gradient flow (Jordan et al. 1998; Santambrogio 2017; Frogner and Poggio 2018). We should emphasize the above derivation is rather formal. Under the suitable assumptions, one can show the existence of $\boldsymbol{\phi}(z, t)$ for the equation (3.3). We refer interested readers to Evans et al. (2005), Ambrosio et al. (2006) and Carrillo and Lisini (2010) for theoretical discussions.

Remark 1 Different choices of energy-dissipation laws lead to different dynamics to an equilibrium. For instance, we can take the energy-dissipation law, as in Liu and Wang (2020b),

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{KL}(\rho(\boldsymbol{x}, t)||\rho^*) = -\int \left(\eta(\rho)||\mathbf{u}||^2 + \nu(\rho)||\nabla\mathbf{u}||^2\right) \mathrm{d}\boldsymbol{x}. \tag{3.5}$$

In this paper, we show that even with the simplest choice of the dissipation $\eta(\rho) = \rho$, the EVI framework can already lead to several new and existing ParVI methods. We will study the benefit of other choices of dissipations in future work.

3.2 Particle-based EVI

In practice, there are two ways to approximate a probability density in $\mathcal{Q}$ defined in (2.1). One is to approximate the transport map $\boldsymbol{\phi}(z, t)$ directly, as used in variational inference with normalizing flow (Rezende and Mohamed 2015). The transport map can be approximated either by a family of parametric transformations (Rezende and Mohamed 2015) or a piece-wise linear map (Carrillo et al. 2018; Liu and Wang 2020a). The main difficulty in such approaches is how to compute $\det[\nabla_z \boldsymbol{\phi}(z, t)]$ efficiently. We refer readers to Rezende and Mohamed (2015), Carrillo et al. (2018), Liu and Wang (2020a) and Papamakarios et al. (2019) for details.

Alternatively, a probability density in $\mathcal{Q}$ can be approximated by an empirical measure defined by a set of sample points $\{\boldsymbol{x}_i(t)\}_{i=1}^N$. As used in many ParVI methods,

$$\rho(\boldsymbol{x}, t) \approx \rho_N(\boldsymbol{x}, t) = \frac{1}{N}\sum_{i=1}^N \delta(\boldsymbol{x} - \boldsymbol{x}_i(t)), \tag{3.6}$$

where $\boldsymbol{x}_i(t) = \boldsymbol{\phi}(\boldsymbol{x}_i(0), t)$ and $\boldsymbol{x}_i(0)$ is sampled from the initial reference distribution ρ_0. The sample points $\{\boldsymbol{x}_i(t)\}_{i=1}^N$ at time t are called "particles" in the ParVIs literature. Instead of computing the map $\boldsymbol{\phi}(z, t)$ explicitly at each time-step, only $\{\boldsymbol{x}_i(t)\}_{i=1}^N$ are computed in ParVIs. One can view this as a deterministic method to sample from the posterior. The evolution of particles $\{\boldsymbol{x}_i(t)\}_{i=1}^N$ can be characterized by a system of ODEs, and it can be derived from the energy-dissipation law (3.1) using the proposed EVI framework, as shown in the follows.

There are two ways to derive such an ODE system. For short, we call them "Approximation-then-Variation" and "Variation-then-Approximation" approaches. Essentially, the two approaches use different orders of density approximation and variational procedure, which may lead to different ODE systems.

The *Approximation-then-Variation* approach starts with a discrete energy-dissipation law

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}_h(\{\boldsymbol{x}_i(t)\}_{i=1}^N) = -2\mathcal{D}_h(\{\boldsymbol{x}_i(t)\}_{i=1}^N, \{\dot{\boldsymbol{x}}_i(t)\}_{i=1}^N), \tag{3.7}$$

which can be obtained by inserting the empirical approximation (3.6) into the continuous energy-dissipation law with a suitable *kernel regularization*. For instance, a discrete version of (3.1), which is the proposed dissipation mechanism of the KL-divergence, can be obtained by applying the particle approximation $\rho_N(\boldsymbol{x}, t)$ to (3.1). To avoid $\ln \delta(\boldsymbol{x} - \boldsymbol{x}_i(t))$ operation, we replace ρ_N by the convolution $K_h * \rho_N$ inside the log function, where K_h is a kernel function. This particle-based approximation leads to the regularized discrete energy-dissipation law

$$\frac{\mathrm{d}}{\mathrm{d}t}\int \rho_N \ln(K_h * \rho_N) + V\rho_N \mathrm{d}\boldsymbol{x} = -\int_\Omega \rho_N ||\mathbf{u}||^2 \mathrm{d}\boldsymbol{x}, \tag{3.8}$$

where

$$K_h * \rho_N = \int K_h(\boldsymbol{x} - \boldsymbol{y})\rho_N(\boldsymbol{y}, t)\mathrm{d}y = \frac{1}{N}\sum_{j=1}^N K_h(\boldsymbol{x} - \boldsymbol{x}_j(t)).$$

We denote $K_h(\boldsymbol{x}-\boldsymbol{x}_j)$ by $K_h(\boldsymbol{x},\boldsymbol{x}_j)$, which is a more conventional notation in the literature. A typical choice of K_h is the Gaussian kernel

$$K_h(\boldsymbol{x}_1,\boldsymbol{x}_2)=\frac{1}{(\sqrt{2\pi}h)^d}\exp\left(-\frac{||\boldsymbol{x}_1-\boldsymbol{x}_2||^2}{h^2}\right). \tag{3.9}$$

The regularized free energy (3.8) is proposed in Carrillo et al. (2019) and has been used to design the Blob variational inference method in Chen et al. (2018). By assuming $\mathbf{u}(\boldsymbol{x}_i(t),t)\approx\dot{\boldsymbol{x}}_i(t)$, the discrete energy is

$$\begin{aligned}&\mathcal{F}_h\left(\{\boldsymbol{x}_i\}_{i=1}^N\right)\\&\quad=\frac{1}{N}\sum_{i=1}^N\left(\ln\left(\frac{1}{N}\sum_{j=1}^N K_h(\boldsymbol{x}_i,\boldsymbol{x}_j)\right)+V(\boldsymbol{x}_i)\right),\end{aligned} \tag{3.10}$$

and the discrete dissipation is

$$-2\mathcal{D}_h\left(\{\boldsymbol{x}_i\}_{i=1}^N,\{\dot{\boldsymbol{x}}_i\}_{i=1}^N\right)=-\frac{1}{N}\sum_{i=1}^N||\dot{\boldsymbol{x}}_i(t))||^2, \tag{3.11}$$

where $\dot{\boldsymbol{x}}_i(t)=\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_i$ is the velocity of each particle.

We can derive the equation of $\dot{\boldsymbol{x}}_i(t)$ via a discrete energetic variational approach (Liu and Wang 2020a)

$$\frac{\delta\mathcal{D}_h}{\delta\dot{\boldsymbol{x}}_i(t)}=-\frac{\delta\mathcal{F}_h}{\delta\boldsymbol{x}_i}, \tag{3.12}$$

which is the energetic variational approach performed at the particle level. An advantage of employing the discrete energetic variational approach is that the resulting system of $\dot{\boldsymbol{x}}_i(t)$'s preserves the variational structure at the particle level. The benefit of this property is discussed in Remark 2 in Sect. 3.3. By direct derivation of the variations of the both sides of (3.12), we obtain a systems of ODEs for $\boldsymbol{x}_i(t)$ as

$$\begin{aligned}\dot{\boldsymbol{x}}_i(t)=-\Bigg(&\frac{\sum_{j=1}^N\nabla_{\boldsymbol{x}_i}K_h(\boldsymbol{x}_i,\boldsymbol{x}_j)}{\sum_{j=1}^N K_h(\boldsymbol{x}_i,\boldsymbol{x}_j)}\\&+\sum_{k=1}^N\frac{\nabla_{\boldsymbol{x}_i}K_h(\boldsymbol{x}_k,\boldsymbol{x}_i)}{\sum_{j=1}^N K_h(\boldsymbol{x}_k,\boldsymbol{x}_j)}+\nabla_{\boldsymbol{x}_i}V(\boldsymbol{x}_i)\Bigg),\\&\text{for } i=1,\ldots,N.\end{aligned} \tag{3.13}$$

It corresponds to the ODE system of the Blob scheme proposed in Chen et al. (2018) for ParVI. However, our derivation of (3.13) is different from Chen et al. (2018).

The *Variation-then-Approximation* approach inserts the empirical approximation (3.6) to (3.3). Note that (3.3) is obtained after the variational step in (2.6). Thus, variation step is done before the approximation step. Formally, the main difficulty in applying the empirical approximation (3.6) to (3.3) is how to evaluate $\nabla\rho_N(\boldsymbol{x},t)$, since $\rho_N(\boldsymbol{x},t)$ is defined by δ functions as in (3.6), and $\nabla\delta$ is not well defined. One way to circumvent this difficulty is to introduce a suitable kernel regularization (Degond and Mustieles 1990; Lacombe and Mas-Gallic 1999). Different kernel regularization methods will result in different ODE systems of particles. In the following, we show that by applying approximation to (3.3), which is the result of variation procedure, we can obtain some existing ParVI methods.

As pointed out in Liu (2017) and Lu et al. (2019), the ODE system corresponding to the standard SVGD is

$$\dot{\boldsymbol{x}}_i(t)=-\sum_{j=1}^N\left(K_h(\boldsymbol{x}_i,\boldsymbol{x}_j)\nabla V(\boldsymbol{x}_j)+\nabla_{\boldsymbol{x}_i}K_h(\boldsymbol{x}_i,\boldsymbol{x}_j)\right).$$

This ODE system can also be obtained using the EVI framework as well. After approximating ρ by ρ_N in (3.3), we can convolute to the right-hand side of (3.3) by a kernel function K_h to obtain

$$\rho_N(\boldsymbol{x},t)\mathbf{u}=K_h*(\rho_N\nabla V+\nabla\rho_N(\boldsymbol{x},t)),$$

which directly leads to the same ODE system as the above one of SVGD.

Another ParVI method is the Gradient Flow with Smoothed test Function (GFSF), proposed by Liu et al. (2019). Using the EVI framework, GFSF can be obtained by applying convolution to both sides of (3.3) with a kernel function K_h

$$K_h*(\rho_N\mathbf{u})=-K_h*(\rho_N\nabla V+\nabla\rho_N),$$

which gives us (let $K_{ij}=K_h(\boldsymbol{x}_i,\boldsymbol{x}_j)$ for short)

$$\sum_{j=1}^N K_{ij}\dot{\boldsymbol{x}}_j(t)=-\sum_{j=1}^N\left(K_{ij}\nabla V(\boldsymbol{x}_j)+\nabla_{\boldsymbol{x}_i}K_h(\boldsymbol{x}_i,\boldsymbol{x}_j)\right).$$

Although its right hand is exactly the descent direction in SVGD, the left is different from SVGD.

The third ParVI method we discuss is the Gradient Flow with Smoothed Density (GFSD), proposed in Lacombe and Mas-Gallic (1999), Degond and Mustieles (1990) and Liu et al. (2019). Under the EVI framework, GFSD can be obtained from $\mathbf{u}=-\nabla\rho/\rho-\nabla V$, by applying convolution to both the numerator and denominator of the first term with a kernel function K_h, i.e.,

$$\mathbf{u}(\boldsymbol{x})=\frac{\rho_N*\nabla K_h}{\rho_N*K_h}-\nabla V(\boldsymbol{x}).$$

It leads to the same ODE system of the GFSD

$$\dot{\boldsymbol{x}}_i(t)=-\left(\frac{\sum_{j=1}^N\nabla_{\boldsymbol{x}_i}K_h(\boldsymbol{x}_i,\boldsymbol{x}_j)}{\sum_{j=1}^N K_h(\boldsymbol{x}_i,\boldsymbol{x}_j)}+\nabla V(\boldsymbol{x}_i)\right).$$

In SVGD and GFSF, kernel function are applied to the velocity equation directly, whereas in GFSD, the δ function in the empirical measure (3.6) is approximated by a suitable kernel $K_h(\boldsymbol{x})$, that is

$$\tilde{\rho}_N(\boldsymbol{x}, t) = \frac{1}{N}\sum_{j=1}^{N} K_h(\boldsymbol{x} - \boldsymbol{x}_j(t)), \tag{3.14}$$

which is more widely used in statistics (Gershman et al. 2012).

Here, we aim to show readers that EVI is a very general framework for variational inference. Even if we choose a simple form of $\eta(\rho) = \rho$, exchanging the variation and approximation steps can lead to various ODE systems of the particles. Some of these ODE systems have already been created from other perspectives as shown above. But many new ParVI methods can be created, as shown in Sect. 3.3. This is an appealing advantage of the proposed EVI.

3.3 Explicit versus implicit Euler

In this subsection, we discuss how to derive a ParVI algorithm from a ODE system. To solve the ODE system (3.13) derived from the "Approximation-then-Variation" approach, one can use the explicit or implicit Euler method. Using the explicit Euler method, we obtain the following numerical scheme

$$\frac{1}{N}\frac{\boldsymbol{x}_i^{n+1} - \boldsymbol{x}_i^n}{\tau_n} = -\frac{\delta \mathcal{F}_h}{\delta \boldsymbol{x}_i}\left(\{\boldsymbol{x}_i^n\}_{i=1}^N\right), \tag{3.15}$$

where τ_n is the step-size. Here, $\mathcal{F}_h$ is the discrete KL-divergence defined in (3.10), and $\frac{\mathcal{F}_h}{\delta \boldsymbol{x}_i}$ is

$$\frac{\delta \mathcal{F}_h}{\delta \boldsymbol{x}_i}\left(\{\boldsymbol{x}_i^n\}_{i=1}^N\right) = \frac{1}{N}\left(\frac{\sum_{j=1}^N \nabla_{\boldsymbol{x}_i} K_h(\boldsymbol{x}_i^n, \boldsymbol{x}_j^n)}{\sum_{j=1}^N K_h(\boldsymbol{x}_i^n, \boldsymbol{x}_j^n)} + \sum_{k=1}^{N} \frac{\nabla_{\boldsymbol{x}_i} K_h(\boldsymbol{x}_k^n, \boldsymbol{x}_i^n)}{\sum_{j=1}^N K_h(\boldsymbol{x}_k^n, \boldsymbol{x}_j^n)} + \nabla_{\boldsymbol{x}_i} V(\boldsymbol{x}_i^n)\right).$$

Scheme (3.15) is exactly the Blob scheme proposed in Chen et al. (2018). The explicit Euler scheme is also used to solve various ODE systems associated with other existing ParVI methods (Liu and Wang 2016; Chen et al. 2018; Liu and Zhu 2018; Liu et al. 2019). To implement these methods, AdaGrad (Duchi et al. 2011) is often used to update the step-size. Although these algorithms perform well in practice, the AdaGrad scales each component of the updating direction differently. As a result, the updating directions of these algorithms are different from their original ODE systems. So the Blob scheme is equivalent to minimize the discrete energy $\mathcal{F}_h(\{\boldsymbol{x}_i\}_{i=1}^N)$ by the AdaGrad algorithm.

An alternative approach is to adopt the implicit Euler scheme for the temporal discretization, i.e.,

$$\frac{1}{N}\frac{\boldsymbol{x}_i^{n+1} - \boldsymbol{x}_i^n}{\tau} = -\frac{\delta \mathcal{F}_h}{\delta \boldsymbol{x}_i}\left(\{\boldsymbol{x}_i^{n+1}\}_{i=1}^N\right). \tag{3.16}$$

The equations (3.16) for $i = 1, \ldots, N$ form a system of nonlinear equations. To solve them, we first define

$$J_n(\{\boldsymbol{x}_i\}_{i=1}^N) := \frac{1}{2\tau}\sum_{i=1}^{N} ||\boldsymbol{x}_i - \boldsymbol{x}_i^n||^2/N + \mathcal{F}_h(\{\boldsymbol{x}_i\}_{i=1}^N). \tag{3.17}$$

In fact, (3.16) is the gradient of $J_n(\{\boldsymbol{x}_i\}_{i=1}^N)$ with respect to the vectorized $\{\boldsymbol{x}_i\}_{i=1}^N$ (see the proof of Theorem 1 in "Appendix D"). Therefore, we can solve the nonlinear equations by solving the optimization problem.

$$\{\boldsymbol{x}_i^{n+1}\}_{i=1}^N = \text{argmin}_{\{\boldsymbol{x}_i\}_{i=1}^N} J_n(\{\boldsymbol{x}_i\}_{i=1}^N), \tag{3.18}$$

which is the celebrated proximal point algorithm (PPA) (Rockafellar 1976). The first term in (3.17) can be viewed as a regularization term. Intuitively, when τ is relatively small, the first term can be the dominating term of $J_n(\{\boldsymbol{x}_i\}_{i=1}^N)$ compared with $\mathcal{F}_h(\{\boldsymbol{x}_i\}_{i=1}^N)$. Since it is also quadratic in $\{\boldsymbol{x}_i\}_{i=1}^N$, it can make the optimization relatively easier to solve than directly minimizing $\mathcal{F}_h(\{\boldsymbol{x}_i\}_{i=1}^N)$. Besides, with a properly chosen τ value, the minimizer of (3.18) can lead to a smaller value of $\mathcal{F}_h(\{\boldsymbol{x}_i\}_{i=1}^N)$, which is also close to $\{\boldsymbol{x}_i^n\}_{i=1}^N$. The optimization problem (3.18) can be solved by a suitable nonlinear optimization. We can show the following convergence result. Its proof is in "Appendix D".

Theorem 1 *For a sufficiently smooth target distribution ρ^* and any given $\{\boldsymbol{x}_i^n\}_{i=1}^N$, there exists at least one minimal solution of (3.18) $\{\boldsymbol{x}_i^{n+1}\}_{i=1}^N$ that also satisfies*

$$\frac{\mathcal{F}_h(\{\boldsymbol{x}_i^{n+1}\}_{i=1}^N) - \mathcal{F}_h(\{\boldsymbol{x}_i^n\}_{i=1}^N)}{\tau} \le -\frac{1}{N}\sum_{i=1}^{N}\frac{||\boldsymbol{x}_i^{n+1} - \boldsymbol{x}_i^n||^2}{2\tau^2}. \tag{3.19}$$

Moreover, if the series $\{\boldsymbol{x}_i^n\}_{i=1}^N$ satisfies (3.19), then $\{\boldsymbol{x}_i^n\}_{i=1}^N$ converges to a stationary point of $\mathcal{F}_h(\{\boldsymbol{x}_i\}_{i=1}^N)$ as $n \to \infty$.

Theorem 1 guarantees the existence of a solution of (3.16) that also decreases the discrete KL-divergence in each iteration. We summarize the algorithm of using the implicit Euler scheme to solve the ODE system (3.13) into Algorithm 1. Here, MaxIter is the maximum number of iteration of the outer loop.

Using Algorithm 1, we update the position of particles by closely following the continuous energy-dissipation law, which provides an efficient way to push the particles to

Algorithm 1 EVI with Implicit Euler Scheme (EVI-Im)

Input: The target distribution $\rho^*(\boldsymbol{x})$ and a set of initial particles $\{\boldsymbol{x}_i^0\}_{i=1}^N$ drawn from a prior $\rho_0(\boldsymbol{x})$.
Output: A set of particles $\{\boldsymbol{x}_i^*\}_{i=1}^N$ approximating ρ^*.
for $n = 0$ **to** MaxIter **do**
  Solve $\{\boldsymbol{x}_i^{n+1}\}_{i=1}^N = \text{argmin}_{\{\boldsymbol{x}_i\}_{i=1}^N} J_n(\{\boldsymbol{x}_i\}_{i=1}^N)$.
  Update $\{\boldsymbol{x}_i^n\}_{i=1}^N$ by $\{\boldsymbol{x}_i^{n+1}\}_{i=1}^N$.
end for

approximate the target distribution. In practice, it is not necessary to obtain the exact minimizer of $J_n(\{\boldsymbol{x}_i\}_{i=1}^N)$ in each iteration. In fact, we only need to find $\{\boldsymbol{x}_i^{n+1}\}_{i=1}^N$ such that

$$\mathcal{F}_h(\{\boldsymbol{x}_i^{n+1}\}_{i=1}^N) \le \mathcal{F}_h(\{\boldsymbol{x}_i^n\}_{i=1}^N),$$

which usually can be achieved in a few steps via the gradient descent method or Newton-like methods with suitable step sizes to $J_n(\{\boldsymbol{x}_i\}_{i=1}^N)$. One can even adopt a line search procedure to guarantee that $J_n(\{\boldsymbol{x}_i^{n+1}\}_{i=1}^N) \le J_n(\{\boldsymbol{x}_i^n\}_{i=1}^N)$. This optimization perspective can also lead to other ParVI methods. For example, the idea of Stein Variational Newton (SVN) type algorithms (Detommaso et al. 2018; Chen et al. 2019) is the same as doing one Newton step to decrease $J_n(\{\boldsymbol{x}_i\}_{i=1}^N)$.

To implement Algorithm 1, we adopt the gradient descent with Barzilai–Borwein step size (Barzilai and Borwein 1988) to solve the optimization problem (3.18). Numerical experiments show that such algorithm usually can find a stationary point of $J_n(\{\boldsymbol{x}\}_{i=1}^N)$ that also satisfies (3.19) with relatively small value of τ. Since it is not necessary to find the exact optimal solution of $J_n(\{\boldsymbol{x}\}_{i=1}^N)$, especially in the early stage of the outer loop in Algorithm 1, we can fix the maximum number of iterations for the inner loop (the loop of minimizing $J_n(\{\boldsymbol{x}\}_{i=1}^N)$) to reduce computation.

Remark 2 The key point in the proposed numerical algorithm is to reformulate the implicit Euler scheme into the optimization problem (3.18), which is equivalent to apply the proximal point algorithm (PPA) (Rockafellar 1976) to the discrete energy $\mathcal{F}_h(\{\boldsymbol{x}_i\}_{i=1}^N)$. We can decrease $\mathcal{F}_h(\{\boldsymbol{x}_i\}_{i=1}^N)$ in each iteration and have the convergence of the algorithm at the discrete level. In other words, in each iteration, the particles are moved as they are intended by the specified dissipation law or the mechanism of decreasing the KL-divergence. This is the benefit of the Variation-then-Approximation approach. For other ParVI methods, it is unclear whether the right-hand sides of the ODEs are the gradients of some functions. Therefore, even though the implicit Euler scheme can be applied to these ParVI methods, the resulted ODE system can not be reformulated as an optimization problem, as we have shown in (3.18).

High-order temporal discretizations can also be used to solve (3.13), such as the Crank–Nicolson and BDF2 schemes (Iserles 2009). Within the variational structure at the particle level, these schemes can also be formulated into optimization problems (Matthes and Plazotta 2019; Du and Feng 2019).

3.4 Choice of kernel

We briefly discuss the choice of kernel, or more precisely, the choice of bandwidth h. The role of kernel function $K_h(\boldsymbol{x}-\boldsymbol{x}_i)$ is essentially to approximate $\delta(\boldsymbol{x}-\boldsymbol{x}_i)$. Considering this role, h should be as small as possible when the number of particles is large. However, in practice, since the number of particles is finite, it is not clear how small h should be. Intuitively, for the Gaussian kernel, h controls the inter-particle distances. In the original SVGD (Liu and Wang 2016), the bandwidth is set to be $h = \text{med}^2/\log N$ where med is the median of the pairwise distance between the current particles. The median trick updates the bandwidth after each iteration. However, as shown in Liu et al. (2019), the median trick only works well for the SVGD. In Liu et al. (2019), the authors proposed a heat equation-based (HE) method. Their idea is to compute the optimal bandwidth after each iteration such that the evolution of approximated density matches the rule of the heat equation. Although the HE method works well during the numerical experiments, it requires solving an optimization problem to obtain the optimal h after each iteration, which is time-consuming. Recently, a matrix-valued kernel for SVGD has been proposed in Wang et al. (2019), in which some anisotropic kernels are used. The selection of kernels is based on the Fisher information, i.e., the Hessian of the $V(\boldsymbol{x})$. Although the matrix-valued kernel works well in practice, as shown in Sect. 4.2, the computational costs will be large.

The optimal bandwidth and the choice of kernel function are problem-dependent. Sometimes, a non-Gaussian kernel might be better (Francois et al. 2005). We do not intend to further the discussion here. In the examples of Sect. 4, we fix the bandwidth of the Gaussian kernel by conducting multiple trials. The results show that a fixed kernel bandwidth works well in many situations for the proposed Algorithm 1.

4 Experiments

We present several examples that demonstrate the proposed EVI scheme summarized in Algorithm 1 (or EVI-Im for short). The results are compared with some other deterministic ParVI methods, including AdaGrad based classical SVGD (Liu and Wang 2016), matrix-valued SVGD (Liu et al. 2019), and Blob method (Chen et al. 2018). Additionally, we also compare our method with a gradient-based MCMC sampling method, Langevin Monte Carlo (LMC) (Rossky et al. 1978; Parisi 1981; Roberts and Tweedie 1996) or its stochastic gradient variant, SGLD (Welling and Teh 2011), given by

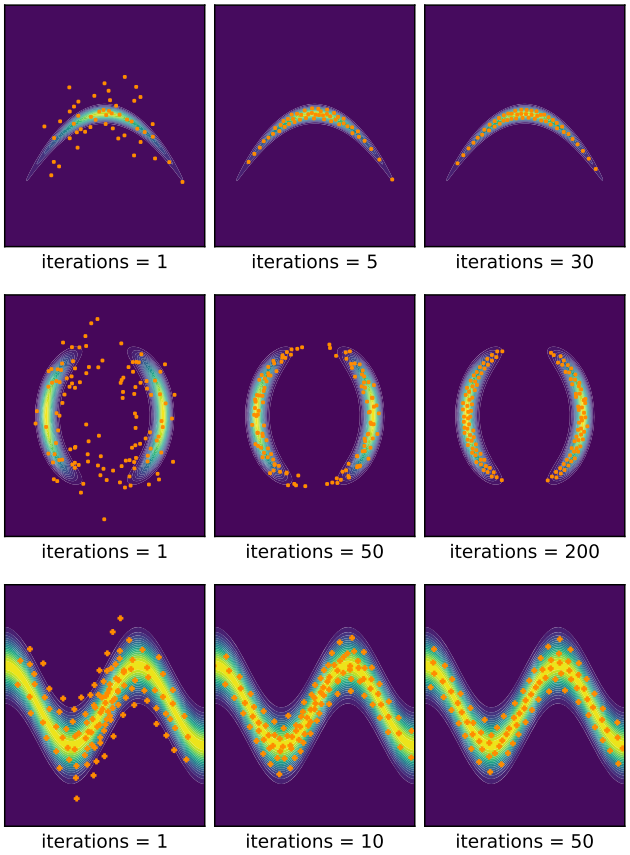

Fig. 2 Particles obtained by EVI-Im algorithm approximating three target distributions plotted as contours

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n - \epsilon_n \nabla(\log \rho^*) + \sqrt{2\epsilon_n}\xi,$$

where ξ is the random term $\xi \sim \mathcal{N}(0, 1)$, and ρ^* is the target/posterior distribution.

In EVI-Im, the number of iterations is "n" defined in the outer loop in Algorithm 1. Therefore, one iteration leads to one update of the positions of *all* the particles. We need to point out that the amount of computation in one iteration of the outer loop of the EVI-Im method is much larger than the other ParVI methods discussed here since the optimization problem (3.18) needs to be solved in each iteration of EVI-Im. To compare the computational costs, we also show the actual CPU time of each method in Sects. 4.2 and 4.4.

4.1 Toy examples via EVI-Im

We first test the EVI-Im on three toy examples, which are widely used as benchmark tests in existing VI literature (Rezende and Mohamed 2015; Chen et al. 2018; Liu et al. 2019). In all three examples, the target distributions are known up to a constant, and we can view the EVI-Im as a deterministic sampling method for these unnormalized probability densities.

The first example is modified from Haario et al. (1999). The target distribution is given by

$$\rho(\boldsymbol{x}) \propto \exp\left\{-\frac{x_1^2}{2} - \frac{1}{2}(10x_2 + 3x_1^2 - 3)^2\right\}.$$

The second example is similar to the examples tested in Rezende and Mohamed (2015) and Liu et al. (2019), and the target distribution is

$$\rho(\boldsymbol{x}) \propto \exp\left\{-2((x_1^2 + x_2^2) - 3)^2 + \log\left(\mathrm{e}^{-2(x_1-2)^2} + \mathrm{e}^{-2(x_2+2)^2}\right)\right\},$$

which has two components. The third example is adapted from Rezende and Mohamed (2015) and Chen et al. (2018) with the target distribution given by

$$\rho(\boldsymbol{x}) \propto \exp\left\{-\frac{1}{2}\left[\frac{x_2 - \sin(\frac{\pi x_1}{2})}{0.4}\right]^2\right\}.$$

In all three examples, the initial particles are sampled from the two-dimensional standard Gaussian distribution. We use $N = 50$ particles for the first example and $N = 120$ particles for the second and third examples. The bandwidth of the kernel is $h = 0.05$ for the first and second examples and $h = 0.2$ for the third example. We set $\tau = 0.01$ for all examples. The final results in Fig. 2 show that the particles returned by the EVI-Im approximate the target distributions reasonably well. The second example is the most challenging one and requires more iterations because the support region (where the density is significantly larger than 0) of the target distribution is not connected and contains two banana-shaped areas.

4.2 Comparison on a star-shaped distribution

The two-dimensional synthesized example studied in Wang et al. (2019) is a challenging one, as the posterior has a star-shaped contour plot shown in Fig. 3. We compare the EVI-Im (set $\tau = 0.5$) with the Blob method (lr $= 0.5$), the classical SVGD (lr $= 0.5$), the matrix-valued SVGD (mixture preconditioning matrix kernel (Wang et al. 2019), lr $= 0.5$), and the LMC method ($\epsilon_n = a(b + n)^{-c}$ with $a = 0.1$, $b = 1$ and $c = 0.55$). The maximum number of iterations of the inner loop in EVI-Im is set to be 100. Here, lr stands for the learning rate. In all five methods, we use $N = 200$ particles and the same initial set of particles sampled from the two-dimensional standard Gaussian distribution. For the EVI-Im and the Blob method, we fix the kernel bandwidth to be $h = 0.1$. The bandwidth matrix in the matrix-valued SVGD is set as the exact Hessian matrices as in Wang et al.

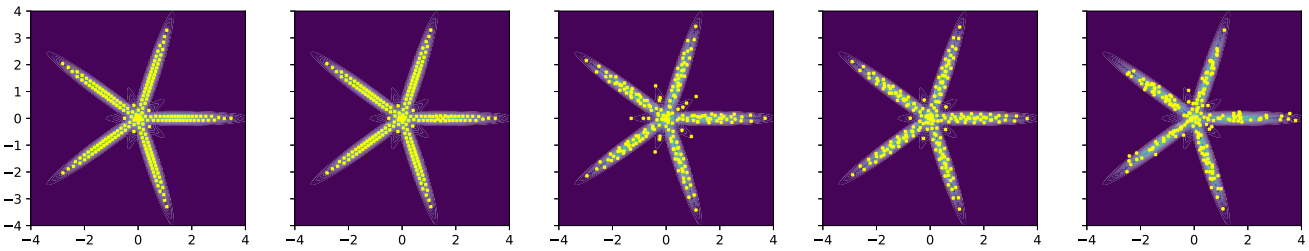

Fig. 3 Particles obtained by various methods [200 particles]: **a** EVI-Im after 20 iterations, **b** Blob method (after 1000 iterations), **c** SVGD (after 1000 iterations), **d** matrix-valued SVGD (after 200 iterations) and **e** LMC (after 3000 iterations)

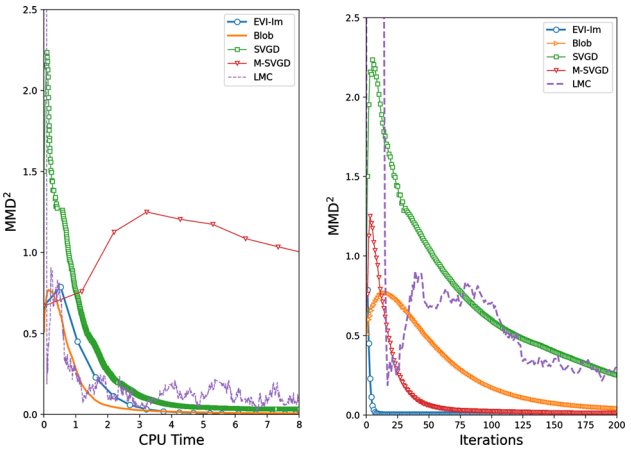

Fig. 4 MMD^2 of each method with respect to CPU time and number of iterations

(2019). To compare the fidelity of the particles to the target distribution, we compute the squared maximum mean discrepancy (MMD^2) defined as Arbel et al. (2019)

$$\text{MMD}^2 = \frac{1}{N^2}\sum_{i,j=1}^{N} k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \frac{1}{M^2}\sum_{i,j=1}^{M} k(\boldsymbol{y}_i, \boldsymbol{y}_j) - \frac{2}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} k(\boldsymbol{x}_i, \boldsymbol{y}_j)$$

with a polynomial kernel $k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\top \boldsymbol{y}/3 + 1)^3$, where $\{\boldsymbol{x}_i\}_{i=1}^N$ are $N = 200$ particles generated by the different methods, and $\{\boldsymbol{y}_j\}_{j=1}^M$ are 5000 samples that generated from ρ^* directly.

The sampling results returned by different methods are shown in Fig. 3. The MMD^2 of each method with respect to the CPU time and the number of iterations is shown in Fig. 4. We observe that the results returned by the EVI-Im and the Blob method are most similar compared to the other methods. As we mentioned earlier, they minimize the same discrete KL-divergence defined in (3.10) with different optimization methods. The EVI-Im uses PPA, whereas the Blob uses AdaGrad. The CPU time for both approaches is also comparable. The particles returned by the EVI-Im, the Blob, and the matrix-valued SVGD appear to align more regularly than those returned by the standard SVGD. Moreover, the particles returned by the three methods are more likely to be concentrated in high probability areas, compared with SVGD and LMC. Using the EVI-Im, we can obtain a good approximation within less than 20 iterations with $\tau = 0.5$. However, since it solves an optimization problem in each iteration, the total CPU time is slightly larger than the Blob method. All methods have similar computational efficiency to the LMC in terms of MMD^2 versus CPU time, except for the matrix-valued SVGD. Its computational cost increases dramatically for computing anisotropic kernels in each iteration. We should emphasize that the total CPU time is sensitive to the choice of learning rate. The learning rate presented here is chosen to have the best performances according to our tests. For EVI-Im, a slightly large time step size τ value is preferred for computational efficiency, but the robustness of the algorithm requires a relatively small τ.

4.3 Mixture model

In this subsection, we consider an example of a simple but interesting mixture model, which is studied in Dai et al. (2016) and Welling and Teh (2011). We sample 1000 observed data from $\boldsymbol{y}_i \sim \frac{1}{2}(N(\omega_1, \sigma^2) + N(\omega_1 + \omega_2, \sigma^2))$, where $(\omega_1, \omega_2) = (1, -2)$ and $\sigma = 2.5$. Using the prior $\omega_1, \omega_2 \sim N(0, 1)$, the posterior distribution is known except the constant, which is the marginal distribution of the data. But it is easy to obtain its two modes, $(1, -2)$ and $(-1, 2)$. The contour plot of the posterior distribution up to the constant is in Fig. 5a.

Figure 5 shows the posterior distribution approximated by EVI-Im and SVGD (lr $= 1.0$). We have tried the SVGD with learning rate lr $= 0.01, 0.1, 0.5, 1.0$ and choose the best learning rate lr $= 1$. For the EVI-Im, we set $\tau = 0.01$. The same $N = 100$ initial particles sampled from the prior are used in both methods, as shown in Fig. 5b. Kernel density estimation with optimal bandwidth selected via cross-validation is used to generate the estimated posterior distribution for both methods. It also shows the approximated distributions of the EVI-Im and the SVGD at different iterations. When

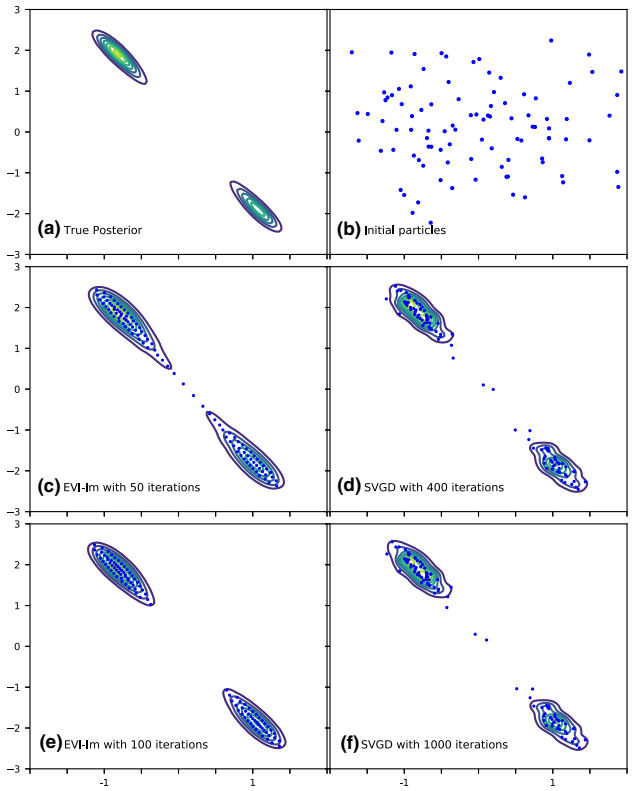

Fig. 5 Comparison of EVI-Im and the classic SVGD (lr = 1) at different stages of iterations

both methods converge, the EVI-Im (100 iterations) approximates the true posterior distribution better than the SVGD (1000 iterations). During the iterations, the particles returned by the EVI-Im also appear to be aligned more regularly. But among the particles returned by the SVGD, some are clustered but some are scattered widely. In Dai et al. (2016), the authors compared many other methods, such as Gibbs sampling, SGLD, and the one-pass sequential Monte Carlo (SMC). Compared to numerical results in Dai et al. (2016), the EVI-Im has better results than Gibbs sampling, SGLD, and SMC, and is also comparable to the particle mirror descent algorithm proposed in Dai et al. (2016).

4.4 Bayesian logistic regression with real data sets

In this subsection, we apply EVI-Im to Bayesian logistic regression models. We first consider a small dataset SPLICE (1000 training entries, 60 features), a benchmark data set used in Mika et al. (1999). Given the data set $\{\boldsymbol{c}_t, y_t\}_{t=1}^{1000}$, the logistic regression model is $p(y_t = 1|\boldsymbol{c}_t, \boldsymbol{\omega}) = [1 + \exp(-\boldsymbol{\omega}^T \boldsymbol{c}_t)]^{-1}$. The unknown parameters $\boldsymbol{\omega}$ are the regression coefficients, whose prior is $N(\boldsymbol{\omega}; \mathbf{0}, \alpha\mathbf{I})$. We compare the EVI-Im method with the classic SVGD and the LMC. We use $N = 20$ particles for each method. The learning rate in SVGD is set to be 0.1. For LMC, we take $\epsilon_n = a(b+n)^{-c}$ with $a = 10^{-4}$, $b = 1$ and $c = 0.55$. For EVI-Im, we take $\tau = 0.01$ and set the maximum number of iteration in the inner loop to be 50. We should emphasize these parameters may not be optimal for all the methods. Figure 6 shows the log-likelihood of the training data and test accuracy for all methods with respect to the CPU time. Although the test accuracies of the three methods are similar and the EVI-Im has a slight advantage, the EVI-Im is shown to achieve a larger log-likelihood with less CPU time of the training data.

We also apply the EVI-Im to a large data set Covertype (Wang et al. 2019), which contains 581,012 data entries and 54 features, and compare the proposed EVI-Im algorithm and the original SVGD method. The prior of the unknown regres-

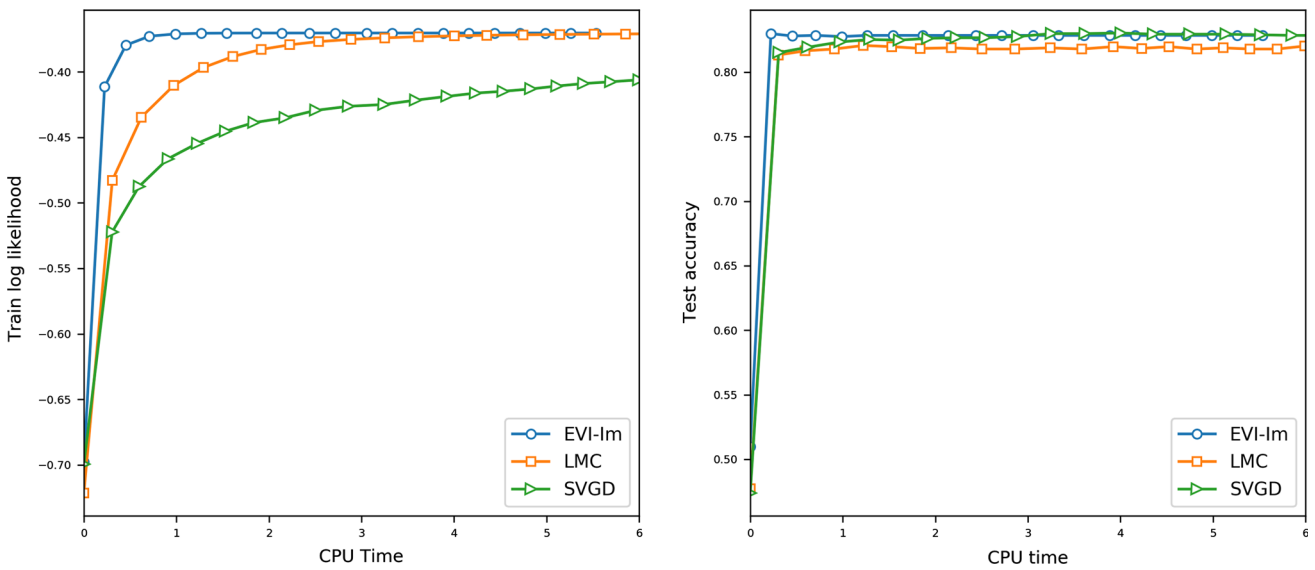

Fig. 6 The log-likelihood of the training data and test accuracy of the SPLICE dataset returned by EVI-Im, SVGD and LMC methods

sion coefficients is chosen to be $p(\boldsymbol{\omega}) = N(\boldsymbol{\omega}; 0, \mathbf{I})$. Due to the large size of the data, the computation of log-likelihood $\nabla \ln \rho^*$ is expensive. Hence, we randomly sample a batch of data to compute a stochastic approximation of $\nabla \ln \rho^*$. The batch size is set to be 256 for all methods. Recall that in the EVI-Im algorithm, we need to solve a minimization problem to update the positions of the particles in each iteration of the outer loop. Since we only estimate $\nabla \ln \rho^*$ using a subset of the complete data, which is only an approximation of the exact estimate using the complete data, the EVI-Im algorithm does not need to achieve the exact local optimality in each iteration. Thus, we choose the stochastic gradient descent method AdaGrad (Duchi et al. 2011) with learning rate lr $= 0.1$ to minimize $J_n(\{\boldsymbol{x}_i\}_{i=1}^N)$. We set the maximum number of iterations for the inner loop of AdaGrad to be 100 in the EVI-Im algorithm. Meanwhile, the time step-size, τ, is set to be 0.1 in the EVI-Im algorithm. For the SVGD method, we choose the best learning rate among lr =0.01, 0.05, 0.1, 0.5, 1.0. For all methods, we use $N = 20$ particles, as in Wang et al. (2019).

In the statistical analysis of real data, it is a common practice to standardize all columns of inputs via their individual mean and standard deviation in the preprocessing stage. Thus, we apply both the EVI-Im and the SVGD algorithms (with lr $= 0.1$) to the standardized data. We also apply the SVGD (with lr $= 1$) to the non-standardized data, which was done in the same way as in Wang et al. (2019). The SVGD is implemented using the codes[1] by Wang et al. (2019). For each method, we have run a total of 20 simulations. In each simulation, we randomly partition the data into training (80% of the whole) and testing (20% of the whole) sets. Figure 7 shows the test accuracy of the classification of EVI-Im and SVGD applied to standardized data and SVGD applied to non-standardized data. The test accuracy is the average of 20 simulations. The CPU time of each point in (b) of Fig. 7 is the average CPU time of 20 simulations of every 100 AdaGrad steps for all three methods under comparison.

For the SVGD (both versions), the number of iterations counts the iterations of the single layer of loop. For the EVI-Im, there are two layers of loops. The outer loop is the for-loop in Algorithm 1 and the inner loop is for the AdaGrad algorithm. As mentioned above, the inner loop of AdaGrad has 100 iterations. To compare it with SVGD, the number of iterations for EVI-Im in Fig. 7a is defined as

$$\text{No. of Outer Iterations} \times 100(\text{No. of Inner Iterations}).$$

Since both methods use the AdaGrad with the same batch size, both methods conduct a very similar amount of computation in each iteration, which is confirmed by the close resemblance between (a) and (b) of Fig. 7. The proposed EVI-Im is the best among the three. We can also compare the EVI-Im algorithm with the matrix-valued SVGD. As shown in Wang et al. (2019), the matrix-valued SVGD can reach an accuracy of 0.75 in less than 500 iterations. Using the EVI-Im algorithm with standardized data, we can reach the same accuracy of 0.75 around 200 iterations.

From Fig. 7, we can first conclude that standardization significantly improves the accuracy and reduce the variance of the SVGD method. This is expected because standardization is essentially applying different bandwidth values to different input dimensions inside the kernel function. As a result, the original SVGD with standardization performs similarly to the matrix-valued SVGD proposed in Wang et al. (2019), although the latter also linearly transforms the SVGD direction by multiplying a preconditioning matrix on the original SVGD direction. For the same reason, EVI-Im algorithm also benefits from standardization, as it is also a kernel-based method.

At last, we point out that the proposed EVI-Im algorithm, the SVGD with or without standardized data, and the matrix-valued SVGD method (Wang et al. 2019) have similar performance when they reach convergence. A major reason is that due to the large size of the data, the KL-divergence is entirely dominated by the log-likelihood. Consequently, the interactions between particles play little effect in the updating of the particles. Thus, there is no significant distinction between different methods when they all reach convergence.

5 Conclusion

In this paper, we introduce a new variational inference framework, called *energetic variational inference* (EVI), in which the procedure of minimizing VI object function is characterized by an *energy-dissipation law*. A VI algorithm can be obtained by employing an energetic variational approach and a proper discretization. The EVI is a general framework. By specifying different components of EVI, we can derive many ParVI algorithms. These components include

- the continuous energy-dissipation law, such as (3.1) and (3.5);
- the order of approximation and variation steps;
- numerical schemes or optimization techniques, such as implicit and explicit Euler, first-order and higher-order temporal discretizations.

We have shown that some combinations of these choices lead to some existing ParVI methods. But many new methods can be created as such. In particular, by using the "Approximation-then-Variation" order, we can derive a particle system that inherits the variational structure from the

[1] Available from https://github.com/dilinwang820/Stein-Variational-Gradient-Descent.

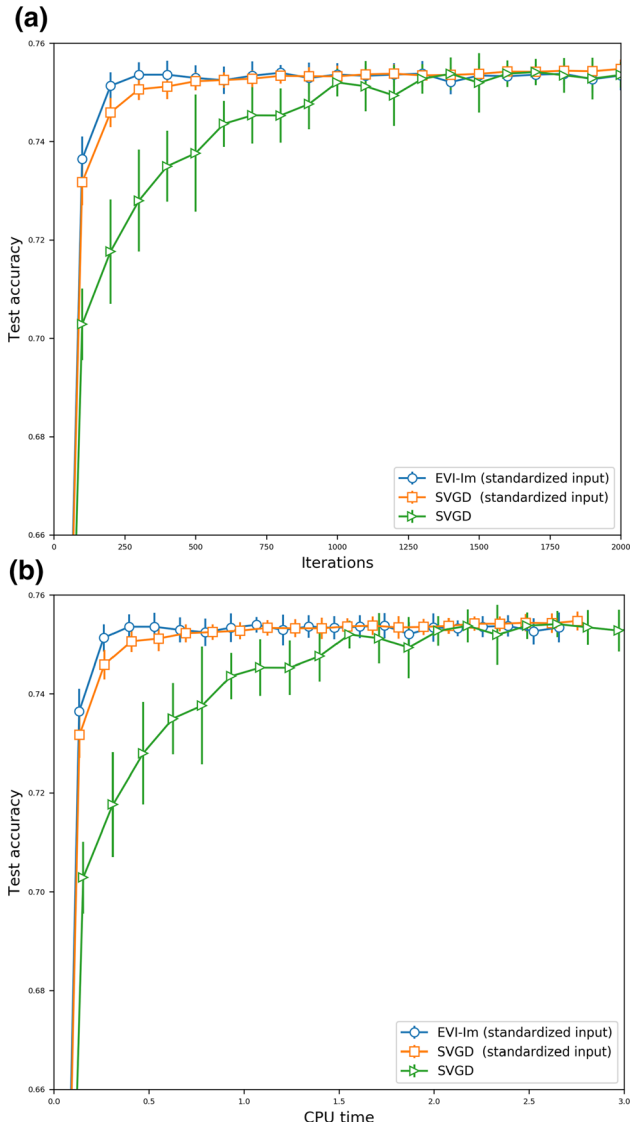

Fig. 7 Test accuracy of 20 simulations for Bayesian logistic regression on Covertype dataset using different methods with respect to **a** the number of iterations and **b** CPU time (in seconds). The number of iterations for EVI-Im is defined as No. of Outer Iterations $\times 100$ (No. of Inner Iterations). The error bar in each curve corresponds to the standard deviation of 20 simulations.

original energy-dissipation law. Numerical examples show that the proposed method has comparable performance with the latest ParVI methods. Another significant aspect is that the EVI framework is not restricted to KL-divergence, and it can be used to minimize other discrepancy measures on the difference between two distributions, such as f-divergence (Ali and Silvey 1966). This opens doors to many varieties in the variational inference literature. The codes and data for all examples are available from Github https://github.com/SimonKafka/EVI.

Acknowledgements Y. Wang and C. Liu are partially supported by the National Science Foundation Grant DMS-1759536. L. Kang is partially supported by the National Science Foundation Grant DMS-1916467.

A Energetic variational approach

In this appendix, we gives a brief introduction to the energetic variational approach. We refer interested readers to Liu (2009) and Giga et al. (2017) for a more comprehensive description.

As mentioned previously, the energetic variational approach provides a paradigm to determine the dynamics of a dissipative system from a prescribed energy-dissipation law, which shifts the main task in the modeling of a dynamic system to the construction of the energy-dissipation law. In physics, an *energy-dissipation law*, yielded by the first and second law of thermodynamics (Giga et al. 2017), is often given by

$$\frac{\mathrm{d}}{\mathrm{d}t}(\mathcal{K}+\mathcal{F})[\boldsymbol{\phi}] = -2\mathcal{D}[\boldsymbol{\phi}, \dot{\boldsymbol{\phi}}], \tag{A.1}$$

where $\boldsymbol{\phi}$ is the state variable, $\mathcal{K}$ is the kinetic energy, $\mathcal{F}$ is the Helmholtz free energy, and $2\mathcal{D}$ is the rate of energy dissipation. If $\mathcal{K} = 0$, one can view (A.1) as a generalization of gradient flow (Hohenberg and Halperin 1977).

The Least Action Principle states that the equation of motion for a Hamiltonian system can be derived from the variation of the action functional $\mathcal{A} = \int_0^T (\mathcal{K} - \mathcal{F})\mathrm{d}\boldsymbol{x}$ with respect to $\boldsymbol{\phi}(\boldsymbol{z}, t)$ (the trajectory), i.e.,

$$\begin{aligned}\delta\mathcal{A} &= \lim_{\epsilon\to 0}\frac{\mathcal{A}[\boldsymbol{\phi}+\epsilon\delta\boldsymbol{\psi}]-\mathcal{A}[\boldsymbol{\phi}]}{\epsilon}\\ &= \int_0^T\int_{\mathcal{X}^t}(f_{\text{inertial}} - f_{\text{conv}})\cdot\delta\boldsymbol{\psi}\mathrm{d}\boldsymbol{x}\mathrm{d}t.\end{aligned}$$

This procedure yields the conservative forces of the system, that is $(f_{\text{inertial}} - f_{\text{conv}}) = \frac{\delta\mathcal{A}}{\delta\boldsymbol{\phi}}$. Meanwhile, according to the MDP, the dissipative force can be obtained by minimizing the dissipation functional with respect to the "rate" $\dot{\boldsymbol{\phi}}$, i.e.,

$$\delta\mathcal{D} = \lim_{\epsilon\to 0}\frac{\mathcal{D}[\dot{\boldsymbol{\phi}}+\epsilon\delta\boldsymbol{\psi}]-\mathcal{D}[\dot{\boldsymbol{\phi}}]}{\epsilon} = \int_{\mathcal{X}^t} f_{\text{diss}}\cdot\delta\psi\mathrm{d}\boldsymbol{x},$$

or $f_{\text{diss}} = \frac{\delta\mathcal{D}}{\delta\dot{\boldsymbol{\phi}}}$. According to the Newton's second law ($F = ma$), we have the force balance condition $f_{\text{inertial}} = f_{\text{conv}} + f_{\text{diss}}$ (f_{inertial} plays role of ma), which defines the dynamics of the system

$$\frac{\delta\mathcal{D}}{\delta\dot{\boldsymbol{\phi}}} = \frac{\delta\mathcal{A}}{\delta\boldsymbol{\phi}}. \tag{A.2}$$

In the case that $\mathcal{K} = 0$, we have

$$\frac{\delta \mathcal{D}}{\delta \dot{\boldsymbol{\phi}}} = -\frac{\delta \mathcal{F}}{\delta \boldsymbol{\phi}}, \tag{A.3}$$

Notice that the free energy is decreasing with respect to the time when $\mathcal{K} = 0$. As an analogy, if we consider VI objective functional as $\mathcal{F}$, (A.1) gives a continuous mechanism to decrease the free energy, and (A.2) or (A.3) gives the equation $\boldsymbol{\phi}(z, t)$.

B Derivation of the transport equation

The transport equation can be derived from the conservation of probability mass directly. Let

$$\mathsf{F}(z, t) = \nabla_z \boldsymbol{\phi}(z, t) \tag{B.1}$$

be the deformation tensor associate with the flow map $\boldsymbol{\phi}(z, t)$, i.e., the Jacobian matrix of $\boldsymbol{\phi}$, then due to the conservation of probability mass, we have

$$\begin{aligned} 0 &= \frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{X}^t} \rho(\boldsymbol{x}, t)\mathrm{d}\boldsymbol{x} = \frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{X}^0} \rho(\boldsymbol{\phi}(z, t), t)\det(\mathsf{F}(z, t))\mathrm{d}z \\ &= \int_{\mathcal{X}^0} \left(\dot{\rho} + \nabla\rho \cdot \mathbf{u} + \rho(\mathsf{F}^{-\mathrm{T}} : \frac{\mathrm{d}\mathsf{F}}{\mathrm{d}t})\right)\det \mathsf{F}\mathrm{d}z \\ &= \int_{\mathcal{X}^t} (\dot{\rho} + \nabla\rho \cdot \mathbf{u} + \rho(\nabla \cdot \mathbf{u}))\,\mathrm{d}\boldsymbol{x} = 0, \end{aligned}$$

which implies that

$$\dot{\rho} + \nabla \cdot (\rho \mathbf{u}) = 0.$$

Here the operation ":" between two matrix, $\mathsf{A} : \mathsf{B} = \sum_i \sum_j A_{ij} B_{ij}$, is the Frobenius inner product between two matrices $\mathsf{A}, \mathsf{B} \in \mathbb{R}^{n\times m}$.

C Computation of Equation (2.10)

In this part, we give a detailed derivation of the variation of $\mathrm{KL}(\rho_{[\boldsymbol{\phi}]}|\rho^*)$ with respect to the flow map $\boldsymbol{\phi}(z) : \mathcal{X}^0 \to \mathcal{X}^t$. Consider a small perturbation of $\boldsymbol{\phi}$

$$\boldsymbol{\phi}^\epsilon(z) := \boldsymbol{\phi}(z) + \epsilon\boldsymbol{\psi}(z),$$

where $\boldsymbol{\psi}(z) = \widetilde{\boldsymbol{\psi}}(\boldsymbol{\phi}(z))$ is a smooth map satisfying

$$\widetilde{\boldsymbol{\psi}} \cdot \boldsymbol{\nu} = 0, \quad \text{on } \partial\mathcal{X}^t$$

with $\boldsymbol{\nu}$ be the outward pointing unit normal on the boundary, $\partial\mathcal{X}^t$. Thus, $\widetilde{\boldsymbol{\psi}} = \boldsymbol{\psi}(\boldsymbol{\phi}^{-1}(\boldsymbol{x}))$ and the above condition indicates that $\widetilde{\boldsymbol{\psi}}$ is diffused to zero at the boundary of $\mathcal{X}^t$. For $\mathcal{X}^0 = \mathcal{X}^d = \mathbb{R}^d$, $\widetilde{\boldsymbol{\psi}} \in C_0^\infty(\mathbb{R}^d)$. We denote F as the Jacobian matrix of $\boldsymbol{\phi}$, i.e., $F_{ij} = \frac{\partial \phi_i}{\partial z_j}$, and F^ϵ is the Jacobian matrix of $\boldsymbol{\phi}^\epsilon$, i.e.,

$$\mathsf{F}^\epsilon := \nabla_z\boldsymbol{\phi} + \epsilon\nabla_z\boldsymbol{\psi}.$$

Then we have

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}\epsilon}\Big|_{\epsilon=0} \mathrm{KL}(\rho_{[\boldsymbol{\phi}^\epsilon]}||\rho^*) \\ &= \frac{\mathrm{d}}{\mathrm{d}\epsilon}\Big|_{\epsilon=0}\left(\int_{\mathcal{X}^0} \frac{\rho_0}{\det(\mathsf{F}^\epsilon)} \ln\left(\frac{\rho_0}{\det(\mathsf{F}^\epsilon)}\right)\det(\mathsf{F}^\epsilon)\mathrm{d}z \right. \\ &\qquad \left. + \int_{\mathcal{X}^0} \frac{\rho_0}{\det(\mathsf{F}^\epsilon)} V(\boldsymbol{\phi}^\epsilon(z))\ \det(\mathsf{F}^\epsilon)\mathrm{d}z\right) \\ &= \frac{\mathrm{d}}{\mathrm{d}\epsilon}\Big|_{\epsilon=0}\left(\int_{\mathcal{X}^0} \rho_0 \ln\rho_0 - \rho_0 \ln\det\mathsf{F}^\epsilon + \rho_0 V(\boldsymbol{\phi}^\epsilon(z))\mathrm{d}z\right) \\ &= \int_{\mathcal{X}^0} -\rho_0 \frac{\mathrm{d}}{\mathrm{d}\epsilon}\Big|_{\epsilon=0}\left(\ln\det(\mathsf{F}^\epsilon) + V(\boldsymbol{\phi}^\epsilon(z))\right)\mathrm{d}z \\ &= \int_{\mathcal{X}^0} -\rho_0(\mathsf{F}^{-\top} : \nabla_z\boldsymbol{\psi}) + (\nabla_{\boldsymbol{x}} V \cdot \boldsymbol{\psi})\rho_0\mathrm{d}z. \end{aligned} \tag{C.1}$$

For two matrices of the same size, define $\mathsf{A} : \mathsf{B} = \sum_i \sum_j A_{ij} B_{ij} = \mathrm{tr}(\mathsf{A}^\top \mathsf{B})$. Since

$$\frac{\mathrm{d}\det(\mathsf{F}^\epsilon)}{\mathrm{d}\epsilon} = \det(\mathsf{F}^\epsilon)\mathrm{tr}\left[(\mathsf{F}^\epsilon)^{-1}\frac{\mathrm{d}\mathsf{F}^\epsilon}{\mathrm{d}\epsilon}\right],$$

we have

$$\begin{aligned} \frac{\mathrm{d}\det(\mathsf{F}^\epsilon)}{\mathrm{d}\epsilon}\Big|_{\epsilon=0} &= \det(\mathsf{F})\mathrm{tr}\left[\mathsf{F}^{-1}\nabla_z\boldsymbol{\psi}\right] \\ &= \det(\mathsf{F})(\mathsf{F}^{-\top} : \nabla_z\boldsymbol{\psi}). \end{aligned}$$

Hence, we have the last result in (C.1).

Based on the definition of $\boldsymbol{\phi}$, we have the following.

$$\begin{aligned} &\boldsymbol{x} = \boldsymbol{\phi}(z), \quad z = \boldsymbol{\phi}^{-1}(\boldsymbol{x}) \\ &\boldsymbol{\psi}(z) = \widetilde{\boldsymbol{\psi}}(\boldsymbol{\phi}(z)) = \widetilde{\boldsymbol{\psi}}(\boldsymbol{x}) \\ &\rho(\boldsymbol{x}) = \rho_0(\boldsymbol{\phi}^{-1}(\boldsymbol{x}))\det(\nabla_{\boldsymbol{x}}\boldsymbol{\phi}^{-1}(\boldsymbol{x})) \\ &\rho_0(z) = \rho_0(\boldsymbol{\phi}^{-1}(\boldsymbol{x})) = \frac{\rho(\boldsymbol{x})}{\det(\nabla_{\boldsymbol{x}}\boldsymbol{\phi}^{-1}(\boldsymbol{x}))}. \end{aligned}$$

The second summand of (C.1) becomes

$$\begin{aligned} &\int_{\mathcal{X}_0} \rho_0\left[(\nabla_{\boldsymbol{x}} V)^\top\boldsymbol{\psi}\right]\mathrm{d}z \\ &= \int_{\mathcal{X}_t} \frac{\rho(\boldsymbol{x})}{\det(\nabla_{\boldsymbol{x}}\boldsymbol{\phi}^{-1}(\boldsymbol{x}))}\left[(\nabla_{\boldsymbol{x}} V)^\top\boldsymbol{\psi}\right]\mathrm{d}\boldsymbol{\phi}^{-1}(\boldsymbol{x}) \\ &= \int_{\mathcal{X}_t} \frac{\rho(\boldsymbol{x})}{\det(\nabla_{\boldsymbol{x}}\boldsymbol{\phi}^{-1}(\boldsymbol{x}))}\left[(\nabla_{\boldsymbol{x}} V)^\top\boldsymbol{\psi}\right]\det(\nabla_{\boldsymbol{x}}\boldsymbol{\phi}^{-1}(\boldsymbol{x}))\mathrm{d}\boldsymbol{x} \\ &= \int_{\mathcal{X}_t} \rho(\boldsymbol{x})\left[(\nabla_{\boldsymbol{x}} V)\cdot\boldsymbol{\psi}\right]\mathrm{d}\boldsymbol{x}. \end{aligned}$$

Now, we investigate the first summand in (C.1). Based on the definition of $\widetilde{\boldsymbol{\psi}}$, we can see that

$$(\nabla_{\boldsymbol{x}}\widetilde{\boldsymbol{\psi}})_{i,j} = \sum_{k=1}^{d} \frac{\partial \psi_i}{\partial z_k} \cdot \frac{\partial z_k}{\partial x_j} = \sum_{k=1}^{d} (\nabla_z \boldsymbol{\psi})_{i,k} (\nabla_{\boldsymbol{x}} \boldsymbol{\phi}^{-1})_{k,j}$$

$$\nabla_{\boldsymbol{x}}\widetilde{\boldsymbol{\psi}} = \nabla_z \boldsymbol{\psi} \left[\nabla_{\boldsymbol{x}} \boldsymbol{\phi}^{-1}\right]^{\top} = (\nabla_z \boldsymbol{\psi}) \mathsf{F}^{-\top},$$

because $\mathsf{F} = \nabla_z \boldsymbol{\phi}(z) = \left(\frac{\partial x_i}{\partial z_j}\right)_{i,j}$, $\mathsf{F}^{-1} = \left(\frac{\partial z_i}{\partial x_j}\right)_{i,j} = (\nabla_{\boldsymbol{x}} \boldsymbol{\phi}^{-1}(\boldsymbol{x}))^{-1}$. Divergence of $\widetilde{\boldsymbol{\psi}}(\boldsymbol{x})$ is

$$\nabla_{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{\psi}}(\boldsymbol{x}) = \sum_{i=1}^{d} \frac{\partial \tilde{\psi}_i}{\partial x_i} = \text{tr}(\text{Jacobian of } \widetilde{\boldsymbol{\psi}})$$
$$= \text{tr}(\nabla_{\boldsymbol{x}} \widetilde{\boldsymbol{\psi}}) = \text{tr}((\nabla_z \boldsymbol{\psi}) \mathsf{F}^{-\top}) = \text{tr}(\mathsf{F}^{-1} \nabla_z \boldsymbol{\psi}).$$

Therefore, the first summand in (C.1) becomes,

$$-\int_{\mathcal{X}_0} \rho_0 (\mathsf{F}^{-\top} : \nabla_z \boldsymbol{\psi}) \mathrm{d}z = -\int_{\mathcal{X}_t} \rho(\boldsymbol{x}) (\nabla_{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{\psi}}) \mathrm{d}\boldsymbol{x}.$$

Following the corollary of divergence theorem,

$$\int_{\mathcal{X}_t} \rho(\boldsymbol{x}) (\nabla_{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{\psi}}) \mathrm{d}\boldsymbol{x} + \int_{\mathcal{X}_t} \widetilde{\boldsymbol{\psi}}^{\top} (\nabla_{\boldsymbol{x}} \rho(\boldsymbol{x})) \mathrm{d}\boldsymbol{x}$$
$$= \oint_{\partial \mathcal{X}_t} \rho(\boldsymbol{x}) (\widetilde{\boldsymbol{\psi}} \cdot \boldsymbol{\nu}) \mathrm{d}S = 0,$$

because the boundary condition $\widetilde{\boldsymbol{\psi}} \cdot \boldsymbol{\nu} = 0$ on $\partial \mathcal{X}_t$. So

$$-\int_{\mathcal{X}_t} \rho(\boldsymbol{x}) (\nabla_{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{\psi}}) \mathrm{d}\boldsymbol{x}$$
$$= \int_{\mathcal{X}_t} \widetilde{\boldsymbol{\psi}}^{\top} (\nabla_{\boldsymbol{x}} \rho(\boldsymbol{x})) \mathrm{d}\boldsymbol{x} = \int_{\mathcal{X}_t} \nabla_{\boldsymbol{x}} \rho(\boldsymbol{x}) \cdot \widetilde{\boldsymbol{\psi}} \mathrm{d}\boldsymbol{x}.$$

Therefore, in $\mathcal{X}^t$, by performing integration by parts, we have

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon}\Big|_{\epsilon=0} \text{KL}(\rho_{[\boldsymbol{\phi}^{\epsilon}]} || \rho^*)$$
$$= \int_{\mathcal{X}^t} -\rho_{[\boldsymbol{\phi}]} (\nabla_{\boldsymbol{x}} \cdot \widetilde{\boldsymbol{\psi}}) + \rho \nabla V \cdot \widetilde{\boldsymbol{\psi}} \mathrm{d}\boldsymbol{x}$$
$$= \int_{\mathcal{X}^t} (\nabla \rho + \rho \nabla V) \cdot \widetilde{\boldsymbol{\psi}} \mathrm{d}\boldsymbol{x},$$

which implies that

$$\frac{\delta \text{KL}(\rho_{[\boldsymbol{\phi}]} || \rho^*)}{\delta \boldsymbol{\phi}} = \nabla \rho + \rho \nabla V \tag{C.2}$$

Recall $V = -\ln \rho^*$. One can notice that if F is an identity matrix, the result in (C.1) can be written as

$$-\mathbb{E}_{z \sim \rho_0} [\text{trace}(\nabla_z \boldsymbol{\psi} + \nabla \ln \rho^* \boldsymbol{\psi}^{\mathrm{T}})],$$

which is exactly the form given by the Stein operator in Liu and Wang (2016).

D Proof of Theorem 1

Proof Let $\mathbf{X} \in \mathbb{R}^D$ be vectorized $\{\boldsymbol{x}_i\}_{i=1}^N$, that is

$$\mathbf{X} = (x_1^{(1)}, \ldots x_N^{(1)}, \ldots x_1^{(d)} \ldots x_N^{(d)}),$$

where $D = N \times d$. Recall that $V(\boldsymbol{x}) = -\ln \rho^*$. For a sufficient smooth target distribution $\rho^*(\boldsymbol{x})$, it is easy to show that

$$\mathcal{F}_h(\{\boldsymbol{x}_i\}_{i=1}^N) = \frac{1}{N} \sum_{i=1}^{N} \left(\ln \left(\frac{1}{N} \sum_{j=1}^{N} K(\boldsymbol{x}_i, \boldsymbol{x}_j) \right) + V(\boldsymbol{x}_i) \right)$$

is continuous, coercive and bounded from below as a function of $\mathbf{X} \in \mathbb{R}^D$. We denote $\mathcal{F}_h(\{\boldsymbol{x}_i\}_{i=1}^N)$ by $\mathcal{F}_h(\mathbf{X})$.

For any given $\{\boldsymbol{x}_i^n\}_{i=1}^N$, recall

$$J_n(\mathbf{X}) = \frac{1}{2\tau} \|\mathbf{X} - \mathbf{X}^n\|^2 + \mathcal{F}_h(\mathbf{X}),$$

where $\|\cdot\|_{\mathbf{X}}^2$ is a norm for $\mathbf{X}$, defined by

$$\|\mathbf{X} - \mathbf{X}^n\|_{\mathbf{X}}^2 = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{x}_i^n\|^2.$$

Since

$$\mathcal{S} = \{J(\mathbf{X}) \leq J(\mathbf{X}^n)\}$$

is a non-empty, bounded, and closed set, by the coerciveness and continuity of $\mathcal{F}_h(\mathbf{X})$, $J_n(\mathbf{X})$ admits a global minimizer $\mathbf{X}^{n+1}$ in $\mathcal{S}$. Since $\mathbf{X}^{n+1}$ is a global minimizer of $J(\mathbf{X})$, we have

$$\frac{1}{2\tau} \|\mathbf{X}^{n+1} - \mathbf{X}^n\|_{\mathbf{X}}^2 + \mathcal{F}_h(\mathbf{X}^{n+1}) \leq \mathcal{F}_h(\mathbf{X}^n),$$

which gives us equation (3.19).

For series $\{\mathbf{X}^n\}$, since

$$\|\mathbf{X}^k - \mathbf{X}^{k-1}\|_{\mathbf{X}}^2 \leq 2\tau (\mathcal{F}_h(\mathbf{X}^{k-1}) - \mathcal{F}_h(\mathbf{X}^k)),$$

we have

$$\sum_{k=1}^{n} \|\mathbf{X}^k - \mathbf{X}^{k-1}\|_{\mathbf{X}}^2 \leq 2\tau (\mathcal{F}_h(\mathbf{X}^0) - \mathcal{F}_h(\mathbf{X}^n)) \leq C,$$

for some constant C that is independent with n. Hence

$$\lim_{n \to \infty} \|\mathbf{X}^n - \mathbf{X}^{n-1}\|_{\mathbf{X}} = 0,$$

which indicates the convergence of $\{\mathbf{X}^n\}$. Moreover, since

$$\mathbf{X}^n = \mathbf{X}^{n-1} - \tau \nabla_{\mathbf{X}} \mathcal{F}_h(\mathbf{X}^n),$$

we have

$$\lim_{n\to\infty} \nabla_{\mathbf{X}} \mathcal{F}_h(\mathbf{X}^n) = 0,$$

so $\{\mathbf{X}^n\}$ converges to a stationary point of $\mathcal{F}_h(\mathbf{X})$. □

References

Ali, S.M., Silvey, S.D.: A general class of coefficients of divergence of one distribution from another. J. R. Stat. Soc. Ser. B **28**(1), 131–142 (1966)

Ambrosio, L., Lisini, S., Savaré, G.: Stability of flows associated to gradient vector fields and convergence of iterated transport maps. Manuscr. Math. **121**(1), 1–50 (2006)

Arbel, M., Korba, A., Salim, A., Gretton, A.: Maximum mean discrepancy gradient flow. In: Advances in Neural Information Processing Systems, pp. 6484–6494 (2019)

Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA J. Numer. Anal. **8**(1), 141–148 (1988)

Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)

Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: a review for statisticians. J. Am. Stat. Assoc. **112**(518), 859–877 (2017)

Carrillo, J.A., Lisini, S.: On the asymptotic behavior of the gradient flow of a polyconvex functional. Nonlinear Partial Differ. Equ. Hyperbolic Wave Phenom. **526**, 37–51 (2010)

Carrillo, J.A., Düring, B., Matthes, D., McCormick, D.S.: A Lagrangian scheme for the solution of nonlinear diffusion equations using moving simplex meshes. J. Sci. Comput. **75**(3), 1463–1499 (2018)

Carrillo, J.A., Craig, K., Patacchini, F.S.: A blob method for diffusion. Calc. Var. Partial. Differ. Equ. **58**(2), 53 (2019)

Casella, G., George, E.I.: Explaining the Gibbs sampler. Am. Stat. **46**(3), 167–174 (1992)

Chen, C., Zhang, R., Wang, W., Li, B., Chen, L.: A unified particle-optimization framework for scalable Bayesian sampling (2018). arXiv preprint arXiv:1805.11659

Chen, P., Wu, K., Chen, J., O'Leary-Roseberry, T., Ghattas, O.: Projected stein variational Newton: a fast and scalable Bayesian inference method in high dimensions (2019). arXiv preprint arXiv:1901.08659

Dai, B., He, N., Dai, H., Song, L.: Provable Bayesian inference via particle mirror descent. In: Artificial Intelligence and Statistics, pp. 985–994 (2016)

Degond, P., Mustieles, F.J.: A deterministic approximation of diffusion equations using particles. SIAM J. Sci. Comput. **11**(2), 293–310 (1990)

Detommaso, G., Cui, T., Marzouk, Y., Spantini, A., Scheichl, R.: A Stein variational Newton method. In: Advances in Neural Information Processing Systems, pp. 9169–9179 (2018)

Du, Q., Feng, X.: The phase field method for geometric moving interfaces and their numerical approximations (2019). arXiv preprint arXiv:1902.04924

Duane, S., Kennedy, A.D., Pendleton, B.J., Roweth, D.: Hybrid Monte Carlo. Phys. Lett. B **195**(2), 216–222 (1987)

Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)

El Moselhy, T.A., Marzouk, Y.M.: Bayesian inference with optimal maps. J. Comput. Phys. **231**(23), 7815–7850 (2012)

Evans, L.C., Savin, O., Gangbo, W.: Diffeomorphisms and nonlinear heat flows. SIAM J. Math. Anal. **37**(3), 737–751 (2005)

Francois, D., Wertz, V., Verleysen, M., et al.: About the locality of kernels in high-dimensional spaces. In: International Symposium on Applied Stochastic Models and Data Analysis, pp. 238–245. Citeseer (2005)

Frogner, C., Poggio, T.: Approximate inference with Wasserstein gradient flows (2018). arXiv preprint arXiv:1806.04542

Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., Rubin, D.B.: Bayesian Data Analysis. Chapman and Hall/CRC, Boca Raton (2013)

Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell. **6**, 721–741 (1984)

Gershman, S.J., Hoffman, M.D., Blei, D.M.: Nonparametric variational inference. In: Proceedings of the 29th International Conference on Machine Learning, pp. 235–242 (2012)

Giga, M.H., Kirshtein, A., Liu, C.: Variational modeling and complex fluids. Handbook of Mathematical Analysis in Mechanics of Viscous Fluids, pp. 1–41 (2017)

Gonzalez, O., Stuart, A.M.: A First Course in Continuum Mechanics. Cambridge University Press, Cambridge (2008)

Haario, H., Saksman, E., Tamminen, J.: Adaptive proposal distribution for random walk Metropolis algorithm. Comput. Stat. **14**(3), 375–396 (1999)

Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**(1), 97–109 (1970)

Hohenberg, P.C., Halperin, B.I.: Theory of dynamic critical phenomena. Rev. Mod. Phys. **49**(3), 435 (1977)

Iserles, A.: A first course in the numerical analysis of differential equations. No. 44 in Cambridge Texts in Applied Mathematics. Cambridge University Press, New York (2009)

Jordan, R., Kinderlehrer, D., Otto, F.: The variational formulation of the Fokker–Planck equation. SIAM J. Math. Anal. **29**(1), 1–17 (1998)

Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Mach. Learn. **37**(2), 183–233 (1999)

Kingma, D.P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., Welling, M.: Improved variational inference with inverse autoregressive flow. In: Advances in Neural Information Processing Systems, pp. 4743–4751 (2016)

Lacombe, G., Mas-Gallic, S.: Presentation and analysis of a diffusion-velocity method. In: ESAIM: Proceedings, vol. 7, pp. 225–233. EDP Sciences (1999)

Li, L., Liu, J.G., Liu, Z., Lu, J.: A stochastic version of Stein variational gradient descent for efficient sampling (2019). arXiv preprint arXiv:1902.03394

Liu, C.: An introduction of elastic complex fluids: an energetic variational approach. In: Multi-Scale Phenomena in Complex Fluids: Modeling, Analysis and Numerical Simulation, pp. 286–337. World Scientific (2009)

Liu, Q.: Stein variational gradient descent as gradient flow. In: Advances in Neural Information Processing Systems, pp. 3115–3123 (2017)

Liu, Q., Wang, D.: Stein variational gradient descent: a general purpose Bayesian inference algorithm. In: Advances in Neural Information Processing Systems, pp. 2378–2386 (2016)

Liu, C., Wang, Y.: On Lagrangian schemes for porous medium type generalized diffusion equations: a discrete energetic variational approach. J. Comput. Phys. **417**, 109566 (2020a)

Liu, C., Wang, Y.: A variational Lagrangian scheme for a phase field model: a discrete energetic variational approach. SIAM J. Sci. Comput. **42**(6), B1541–B1569 (2020b)

Liu, C., Zhu, J.: Riemannian Stein variational gradient descent for Bayesian inference. In: 32nd AAAI Conference on Artificial Intelligence (2018)
Liu, C., Zhuo, J., Cheng, P., Zhang, R., Zhu, J.: Understanding and accelerating particle-based variational inference. In: International Conference on Machine Learning, pp. 4082–4092 (2019)
Lu, J., Lu, Y., Nolen, J.: Scaling limit of the Stein variational gradient descent: the mean field regime. SIAM J. Math. Anal. **51**(2), 648–671 (2019)
MacKay, D.J., Mac Kay, D.J.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge (2003)
Matthes, D., Plazotta, S.: A variational formulation of the BDF2 method for metric gradient flows. ESAIM: Math. Model. Numer. Anal. **53**(1), 145–172 (2019)
Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. J. Chem. Phys. **21**(6), 1087–1092 (1953)
Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K.R.: Fisher discriminant analysis with kernels. In: Neural networks for signal processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop, pp. 41–48. IEEE (1999)
Murphy, K.P.: Machine Learning: A Probabilistic Perspective. MIT Press, Cambridge (2012)
Neal, R.M.: Probabilistic inference using Markov chain Monte Carlo methods. Department of Computer Science, University of Toronto Toronto, Ontario, Canada (1993)
Neal, R.M., Hinton, G.E.: A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Learning in Graphical Models, pp. 355–368. Springer (1998)
Onsager, L.: Reciprocal relations in irreversible processes. I. Phys. Rev. **37**(4), 405 (1931a)
Onsager, L.: Reciprocal relations in irreversible processes. II. Phys. Rev. **38**(12), 2265 (1931b)
Papamakarios, G., Nalisnick, E., Rezende, D.J., Mohamed, S., Lakshminarayanan, B.: Normalizing flows for probabilistic modeling and inference (2019). arXiv preprint arXiv:1912.02762
Parisi, G.: Correlation functions and computer simulations. Nucl. Phys. B **180**(3), 378–384 (1981)
Rayleigh, L.: Note on the numerical calculation of the roots of fluctuating functions. Proc. Lond. Math. Soc. **1**(1), 119–124 (1873)
Rezende, D.J., Mohamed, S.: Variational inference with normalizing flows (2015). arXiv preprint arXiv:1505.05770
Roberts, G.O., Tweedie, R.L., et al.: Exponential convergence of Langevin distributions and their discrete approximations. Bernoulli **2**(4), 341–363 (1996)
Rockafellar, R.T.: Monotone operators and the proximal point algorithm. SIAM J. Control Optim. **14**(5), 877–898 (1976)
Rossky, P.J., Doll, J.D., Friedman, H.L.: Brownian dynamics as smart Monte Carlo simulation. J. Chem. Phys. **69**(10), 4628–4633 (1978)
Salimans, T., Kingma, D., Welling, M.: Markov chain Monte Carlo and variational inference: bridging the gap. In: International Conference on Machine Learning, pp. 1218–1226 (2015)
Salman, H., Yadollahpour, P., Fletcher, T., Batmanghelich, K.: Deep diffeomorphic normalizing flows (2018). arXiv preprint arXiv:1810.03256
Santambrogio, F.: {Euclidean, metric, and Wasserstein} gradient flows: an overview. Bull. Math. Sci **7**(1), 87–154 (2017)
Sonoda, S., Murata, N.: Transport analysis of infinitely deep neural network. J. Mach. Learn. Res. **20**(1), 31–82 (2019)
Stuart, A.M.: Inverse problems: a Bayesian perspective. Acta Numer. **19**, 451–559 (2010)
Tabak, E.G., Vanden-Eijnden, E., et al.: Density estimation by dual ascent of the log-likelihood. Commun. Math. Sci. **8**(1), 217–233 (2010)
Temam, R., Miranville, A.: Mathematical Modeling in Continuum Mechanics. Cambridge University Press, Cambridge (2005)
Villani, C.: Optimal Transport: Old and New, vol. 338. Springer, Berlin (2008)
Wainwright, M.J., Jordan, M.I., et al.: Graphical models, exponential families, and variational inference. Found. Trends® Mach. Learn. **1**(1–2), 1–305 (2008)
Wang, D., Tang, Z., Bajaj, C., Liu, Q.: Stein variational gradient descent with matrix-valued kernels. In: Advances in Neural Information Processing Systems, pp. 7834–7844 (2019)
Welling, M., Teh, Y.W.: Bayesian learning via stochastic gradient Langevin dynamics. In: Proceedings of the 28th International Conference on Machine Learning, pp. 681–688 (2011)