DPVI: A Dynamic-Weight Particle-Based Variational Inference Framework

Chao Zhang[1,2], **Zhijian Li**[3*], **Xin Du**[3†] and **Hui Qian**[1,2]

[1]College of Computer Science and Technology, Zhejiang University
[2]Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies
[3]Information Science and Electronic Engineering, Zhejiang University
{zczju, lizhijian, duxin, qianhui}@zju.edu.cn

Abstract

The recently developed Particle-based Variational Inference (ParVI) methods drive the empirical distribution of a set of fixed-weight particles towards a given target distribution by iteratively updating particles' positions. However, the fixed weight restriction greatly confines the empirical distribution's approximation ability, especially when the particle number is limited. In this paper, we propose to dynamically adjust particles' weights according to a Fisher-Rao reaction flow. We develop a general Dynamic-weight Particle-based Variational Inference (DPVI) framework according to a novel continuous composite flow, which evolves the positions and weights of particles simultaneously. We show that the mean-field limit of our composite flow is actually a Wasserstein-Fisher-Rao gradient flow of the associated dissimilarity functional. By using different finite-particle approximations in our general framework, we derive several efficient DPVI algorithms. The empirical results demonstrate the superiority of our derived DPVI algorithms over their fixed-weight counterparts.

1 Introduction

Recently, Particle-based Variational Inference (ParVI) methods have drawn much attention in the Bayesian inference literature, due to their success in efficiently approximating the target posterior distribution π [Liu and Wang, 2016; Liu and Zhu, 2018; Liu and Wang, 2018; Pu *et al.*, 2017; Zhu *et al.*, 2020]. The core of ParVIs lies at evolving the empirical distribution of M *fixed-weight* particles by simulating a *continuity equation* through its easy-to-calculate finite-particle position transport approximation [Liu *et al.*, 2019a]. Typically, the continuity equation is constructed according to the Wasserstein gradient flow of certain dissimilarity functional $\mathcal{F}(\mu) := \mathcal{D}(\mu|\pi)$ vanishing at $\mu = \pi$ [Liu *et al.*, 2019a]. By using different dissimilarity functionals $\mathcal{F}$ and position transport approximations, several ParVIs have been proposed, e.g., Stein Variational Gradient Descent (SVGD) method [Liu and Wang, 2016], Blob[Chen *et al.*, 2018a] GFSD [Liu *et al.*, 2019a], the Kernel Stein Discrepancy Descent (KSDD) [Korba *et al.*, 2021].

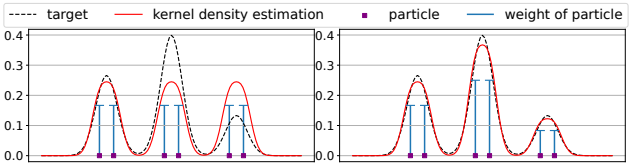

Figure 1: Approximating a Gaussian mixture distribution with six particles. The black dashed lines denote the target density, the solid red lines are the densities of particles (estimated using the kernel density estimator), and the heights of the solid blue lines represent the weight of each particle. The left sub-figure shows the result of the fixed-weight ParVI algorithm Blob, and the right sub-figure is from our dynamic-weight D-Blob-CA algorithm.

Fixed weight restriction. Existing ParVIs have a common fixed weight restriction, i.e., they all keep the particles' weights fixed during the whole procedure, and only update the positions of particles according to the position transport approximation derived from a continuity equation. This restriction severely confines the empirical distribution's approximation ability, especially when the particle number M is limited (depicted in Figure 1). To mitigate the influence of this restriction, existing ParVIs require plenty of particles to obtain satisfying approximation accuracy [Liu and Wang, 2018; Korba *et al.*, 2020]. As a result, a large amount of computation is usually needed since the per-iteration computational cost in ParVIs is typically in the square order of M. Actually, a huge deviation between the empirical distribution and the target π is often observed when the particle number is insufficient due to a limited computational budget [Zhang *et al.*, 2020a; Zhang *et al.*, 2020b].

Thus, it is in great need to find an effective weight adjustment strategy and design dynamic-weight ParVI algorithms which could achieve a high approximation accuracy with fewer particles and hence less computation. Note that, though the continuity equation underlying existing fixed-weight ParVIs can be directly transformed into an easy-to-calculate position transport approximation, adjusting weights according to the continuity equation is generally infeasible as the divergence operator in it would introduce great computa-

*Chao Zhang and Zhijian Li contribute equally.
†Contact Author.

tional challenge. Constructing effective algorithms to evolve a set of dynamic-weight particles towards the target π is still an open problem in the ParVI field.

To tackle this problem, we propose to dynamically adjust particles' weights according to the Fisher-Rao reaction flow of the underlying dissimilarity $\mathcal{F}$, and design a continuous-time composite flow, which evolves the positions and weights of M particles simultaneously. Specifically, the composite flow is a combination of a finite-particle approximation of the reaction flow and a finite-particle position transport approximation of the continuity equation. Different dynamic-weight ParVI algorithms can be obtained by discretizing the continuous flow with different discretization schemes and dissimilarities $\mathcal{F}$. The contribution of our paper are listed as follows:

- We show that the mean-field limit of our proposed composite flow is actually the gradient flow of $\mathcal{F}$ in the Wasserstein-Fisher-Rao space, which leads to an extra decrease of $\mathcal{F}$ compared with both the Wasserstein gradient flow underlying existing fixed-weight ParVI algorithms and the Fisher-Rao reaction flow.
- We propose a general Dynamic-weight Particle-based Variational Inference (DPVI) framework, which utilizes an Euler discretization of the composite flow and adopts a Gauss-Siedel-type strategy to update the positions and the weights. Note that the weight adjustment step can be implemented without bringing much extra computation compared to the position update step. By adopting different dissimilarities $\mathcal{F}$ in DPVI, we can obtain different efficient dynamic-weight ParVI algorithms.
- We propose three efficient DPVI algorithms by using different dissimilarities $\mathcal{F}$ and their associated finite-particle approximations in the general framework. Besides, we also derive three duplicate/kill variants of our proposed algorithms, where a probabilistic discretization to the weight adjustment part in the composite flow is used to dynamically duplicate/kill particles, instead of adjusting the particles' weights continuously.

We evaluate our algorithms on various synthetic and real-world tasks. The empirical results demonstrate the superiority of our dynamic weight strategy over the fixed weight strategy, and our DPVI algorithms constantly outperform their fixed-weight counterparts in all the tasks.

Notation. Given a probability measure μ on $\mathbb{R}^d$, we denote $\mu \in \mathcal{P}_2(\mathbb{R}^d)$ if its second moment is finite. For a given functional $\mathcal{F}(\mu) : \mathcal{P}_2(\mathbb{R}^d) \to \mathbb{R}$, $\frac{\delta \mathcal{F}(\rho)}{\delta \rho} : \mathbb{R}^d \to \mathbb{R}$ denote its first variation at $\mu = \rho$. Besides, we use ∇ and $\nabla \cdot ()$ to denote the gradient and the divergence operator, respectively.

2 Preliminaries

2.1 Particle-Based Variational Inference Methods

When dealing with Bayesian inference tasks, classical Variational Inference methods approximate the target posterior π with an easy-to-sample distribution μ, and recast the inference task as an optimization problem over $\mathcal{P}_2(\mathbb{R}^d)$ (or its subspace) [Ranganath *et al.*, 2014]:

$$\min_{\mu \in \mathcal{P}_2(\mathbb{R}^d)} \mathcal{F}(\mu) := \mathcal{D}(\mu|\pi). \tag{1}$$

To solve this optimization problem, one can consider a descent flow of $\mathcal{F}(\mu)$ in the Wasserstein space, which transports any initial distribution μ_0 towards the target π [Wibisono, 2018]. Specifically, the descent flow of $\mathcal{F}(\mu)$ is described by the following *continuity equation* [Ambrosio *et al.*, 2008]:

$$\partial_t \mu_t = -\nabla \cdot (\mu_t \mathbf{v}_{\mu_t}), \tag{2}$$

where $\mathbf{v}_{\mu_t} : \mathbb{R}^d \to \mathbb{R}^d$ is a vector field that defines the direction of position transportation. To ensure a descent of $\mathcal{F}(\mu_t)$ over time t, the vector field $\mathbf{v}_{\mu_t}$ should satisfy the following inequality [Ambrosio *et al.*, 2008]:

$$\frac{\mathrm{d}\mathcal{F}(\mu_t)}{\mathrm{d}t} = \int \langle \nabla \frac{\delta \mathcal{F}(\mu_t)}{\delta \mu_t}, \mathbf{v}_{\mu_t} \rangle \mathrm{d}\mu_t \leq 0. \tag{3}$$

A straightforward choice of $\mathbf{v}_{\mu_t}$ is $\mathbf{v}_{\mu_t} = -\nabla \frac{\delta \mathcal{F}(\mu_t)}{\delta \mu_t}$, which is actually the steepest descent direction of $\mathcal{F}(\mu_t)$. From now on, we fix this choice of $\mathbf{v}_{\mu_t}$ in (2). We note that the continuity equation (2) with this $\mathbf{v}_{\mu_t}$ is also known as the *Wasserstein gradient flow* of $\mathcal{F}$.

To simulate the Wasserstein gradient flow of $\mathcal{F}$, existing ParVIs evolve a set of M fixed-weight particles and use the empirical distribution $\tilde{\mu}_t = \sum_{i=1}^M a_t^i \delta_{\mathbf{x}_t^i}$ to approximate μ_t in (2), where $\mathbf{x}_t^i$ and a_t^i (usually set to $1/M$) denote the position and the weight of the i-th particle at time t, respectively. Specifically, ParVIs update the position of each particle $\mathbf{x}_t^i$ according to the following finite-particle *position transport approximation* of the continuity equation (2) [Chen *et al.*, 2018a; Liu, 2017; Craig and Bertozzi, 2016]:

$$\mathrm{d}\mathbf{x}_t^i = \mathbf{v}_{\tilde{\mu}_t}(\mathbf{x}_t^i)\mathrm{d}t, \tag{4}$$

where $\mathbf{v}_{\tilde{\mu}_t}$ is an approximation of $\mathbf{v}_{\mu_t}$ through the empirical distribution $\tilde{\mu}_t$. It can be verified that the empirical distribution $\tilde{\mu}_t$ weakly converges to μ_t defined in (2) when $M \to \infty$ under mild conditions [Korba *et al.*, 2020; Liu, 2017; Liu and Wang, 2018]. Therefore, one can obtain different ParVIs by choosing proper $\mathbf{v}_{\tilde{\mu}_t}$ and discretizing (4) with certain scheme (the first-order explicit Euler discretization is set as a default). Note that existing ParVIs only update the position $\mathbf{x}_t^i$ and keep the weight a_t^i fixed during the whole procedure, since adjusting weight a_t^i according to (2) involves the calculation of the second-order derivative of the first variation due to the divergence operator, which is usually expensive.

To develop a ParVI method, it remains to select a proper dissimilarity functional $\mathcal{F}$ and construct an approximation $\mathbf{v}_{\tilde{\mu}_t}$ of the vector field $\mathbf{v}_{\mu_t}$. From the seminal SVGD to the subsequent Blob and GFSD, KL-divergence is widely adopted as the underlying dissimilarity functional [Liu and Wang, 2016; Chen and Zhang, 2017; Chen *et al.*, 2018a; Liu *et al.*, 2019a; Liu *et al.*, 2019b; Zhang *et al.*, 2020a]. The associated vector field is defined as follows [Jordan *et al.*, 1998; Liu *et al.*, 2019a]:

$$\mathbf{v}_{\mu_t} = -\nabla \frac{\delta \mathrm{KL}(\mu_t|\pi)}{\delta \mu_t} = -\nabla \log \frac{\mu_t}{\pi} = \nabla \log \pi - \nabla \log \mu_t.$$

As $\nabla \log \mu_t$ is undefined with the empirical distribution $\tilde{\mu}_t$, the KL-divergence based ParVIs use different approaches to construct suitable approximations to the vector field. In SVGD, Liu and Wang [2016] restrict the vector field $\mathbf{v}_{\mu_t}$ within the unit ball of a Reproducing Kernel Hilbert Space (RKHS), and propose to approximate $\mathbf{v}_{\mu_t}$ by

$$\mathbf{v}_{\tilde{\mu}_t}(\cdot) = \mathbb{E}_{\mathbf{x}' \sim \tilde{\mu}_t} \left[K(\mathbf{x}', \cdot) \log \pi(\mathbf{x}) + \nabla_{\mathbf{x}'} K(\mathbf{x}', \cdot) \right], \quad (5)$$

where K is a kernel function, such as the Radial Basis Function (RBF) kernel. Subsequently, Blob [Chen *et al.*, 2018a] reformulates the intractable term $\nabla \log \mu_t$ by partly smoothing the density with a kernel function K, while GFSD [Liu *et al.*, 2019a] directly approximates μ_t by $\tilde{\mu}_t * K$, where $*$ denotes the convolution operator. Recently, Kernel Stein Discrepancy Descent (KSDD) [Korba *et al.*, 2021] method considers the Kernel Stein Discrepancy (KSD) [Liu *et al.*, 2016] as the dissimilarity $\mathcal{F}$, whose first variation is compatible with empirical distribution and can be calculated directly.

2.2 Fisher-Rao Distance and Reaction Flow

The Fisher-Rao distance [Rao, 1945; Kakutani, 1948] is a metric defined for general positive Radon measures and allows comparing measures with mass variations. The positive Radon measure space equipped with the Fisher-Rao distance is known as the Fisher-Rao space. For a given dissimilarity functional $\mathcal{F}(\mu)$, its gradient flow in the Fisher-Rao space, also named as the Fisher-Rao reaction flow [Gallouët and Monsaingeon, 2017], is described by the following equation [Wang and Li, 2019; Liero *et al.*, 2016]:

$$\partial_t \mu_t = -\alpha_{\mu_t} \mu_t, \ \alpha_{\mu_t} = \frac{\delta \mathcal{F}(\mu_t)}{\delta \mu_t} - \int \frac{\delta \mathcal{F}(\mu_t)}{\delta \mu_t} \mathrm{d}\mu_t, \quad (6)$$

where $\alpha_{\mu_t} : \mathbb{R}^d \to \mathbb{R}$ represents a construction/destruction function of mass. Since the average variation of μ_t equals zero due to the second integral term, the total mass of μ_t is conserved during the whole procedure [Lu *et al.*, 2019; Rotskoff *et al.*, 2019]. The target distribution π is actually an invariant distribution of this flow, as the first variation of the dissimilarity functional $\mathcal{F}$ vanishes at π, i.e., $\frac{\delta \mathcal{F}(\pi)}{\delta \pi} = 0$. It can be verified that with a proper $\mathcal{F}$, the process (6) starting from a given μ_0 evolves towards the target distribution π [Kondratyev *et al.*, 2016].

For a fixed position $\mathbf{x}$, the process (6) provides an effective way to adjust its density (weight) $\mu_t(\mathbf{x})$ at each time t according to the function α_{μ_t}. Thus, given a set of weighted particles and its empirical distribution $\tilde{\mu}_t$, one can adjust the weight by discretizing the reaction flow with an empirical approximate construction/deconstruction function $\alpha_{\tilde{\mu}_t}$. Though the reaction flow has been adopted in particle-based methods in other literature, such as MCMC [Lu *et al.*, 2019], global minimization [Rotskoff *et al.*, 2019] and generative models [Mroueh and Rigotti, 2020], to the best of our knowledge, it has never been adopted in the ParVI literature to adjust the weights of particles.

3 Methodology

In this section, we first construct a composite flow that evolves positions and weights of particles simultaneously and investigate its mean-field property. Then, we develop our DPVI framework by discretizing this composite flow. We finally provide three effective DPVI algorithms by using different dissimilarity functionals $\mathcal{F}$ (KL-divergence and KSD) and finite-particle approximations. Besides, we also derive the duplicate/kill variants of our proposed algorithms.

3.1 Continuous-Time Composite Flow

Based on the position transport approximation (4) for displacing the position and the Fisher-Rao reaction flow (6) for adjusting weight in previous sections, we consider the following composite flow that evolves the positions $\mathbf{x}^i$'s and the weights a^i's of M particles simultaneously.

$$\begin{cases} \mathrm{d}\mathbf{x}_t^i = \mathbf{v}_{\tilde{\mu}_t} \mathrm{d}t, \\ \mathrm{d}a_t^i = -\left(U_{\tilde{\mu}_t}(\mathbf{x}_t^i) - \sum_{i=1}^M a_t^i U_{\tilde{\mu}_t}(\mathbf{x}_t^i) \right) a_t^i \mathrm{d}t, \\ \tilde{\mu}_t = \sum_{i=1}^M a_t^i \delta_{\mathbf{x}_t^i}. \end{cases} \quad (7)$$

For ease of notation, we use U_μ and $\mathbf{v}_\mu$ to denote the first variation of $\mathcal{F}(\mu)$ at μ and the vector field associated with it, respectively, i.e., $U_\mu = \frac{\delta \mathcal{F}(\mu)}{\delta \mu}$ and $\mathbf{v}_\mu = -\nabla U_\mu$. Although the mean-field limit of the empirical distribution with either the position update part $\mathrm{d}\mathbf{x}_t^i$ or the weight adjustment part $\mathrm{d}a_t^i$ alone has the target distribution π as its stationary distribution, the behaviour of the empirical distribution $\tilde{\mu}_t$ in (7) remains unknown. Thus, we first investigate the mean-field limit of $\tilde{\mu}_t$ and show that it actually follows the gradient flow of $\mathcal{F}$ in the Wasserstein-Fisher-Rao space when $M \to \infty$.

Proposition 1. *Suppose the empirical distribution $\tilde{\mu}_0^M$ of M weighted particles weakly converges to a distribution μ_0 when $M \to \infty$. Then, the path of (7) starting from $\tilde{\mu}_0^M$ weakly converges to a solution of the following partial differential equation starting from μ_0 as $M \to \infty$:*

$$\partial_t \mu_t = \nabla \cdot (\mu_t \nabla U_{\mu_t}) - \left(U_{\mu_t} - \int U_{\mu_t} \mathrm{d}\mu_t \right) \mu_t, \quad (8)$$

which is actually the gradient flow of $\mathcal{F}$ in the Wasserstein-Fisher-Rao space.

Compared to the Wasserstein gradient flow (2) of $\mathcal{F}$, the Wasserstein-Fisher-Rao gradient flow (8) has an additional density adjustment part (second term in the r.h.s. of equation (8)). Due to the limit of space, we defer the discussion of the descending property of (8) in our long version. Actually, it can be verified that (8) results in an extra decreasing term than the Wasserstein gradient flow used in the fixed-weight ParVIs due to the additional density adjustment part.

3.2 Dynamic-Weight ParVI Framework

Generally, it is impossible to obtain an analytic solution of the continuous composite flow (7), thus a numerical integration method is required to derive an approximate solution. Note that any numerical solver, such as the implicit Euler method [Platen and Bruti-Liberati, 2010] and higher-order Runge-Kutta method [Butcher, 1964] can be used. Here, we adopt the first-order explicit Euler discretization [Süli and Mayers, 2003] since it is simple and easy-to-implement, and

propose our Dynamic-weight Particle-based Variational Inference (DPVI) framework, as listed in Algorithm 1.

Starting from M weighted particles located at $\{\mathbf{x}_0^i\}_{i=1}^M$ with weights $\{a_0^i\}_{i=0}^M$, DPVI first updates the positions of particles according to the following rule:

$$\mathbf{x}_{k+1}^i = \mathbf{x}_k^i + \eta \mathbf{v}_{\tilde{\mu}_k}(\mathbf{x}_k^i), \tag{9}$$

where $\tilde{\mu}_k = \sum_{i=1}^M a_k^i \delta_{\mathbf{x}_k^i}$. Then, it adjusts the particles' weights following

$$a_{k+1}^i = a_k^i - \lambda \eta \bar{U}_{\tilde{\mu}_{k+1/2}}(\mathbf{x}_{k+1}^i) a_k^i, \tag{10}$$

where $\bar{U}_{\tilde{\mu}_{k+1/2}} = U_{\tilde{\mu}_{k+1/2}} - \sum_{i=1}^M a_k^i U_{\tilde{\mu}_{k+1/2}}(\mathbf{x}_{k+1}^i)$, and $\tilde{\mu}_{k+1/2} = \sum_{i=1}^M a_k^i \delta_{\mathbf{x}_{k+1}^i}$ represents the empirical distribution after the position update (9). Here, we assume that a suitable empirical approximation $U_{\tilde{\mu}_k}$ of the first variation is already constructed, and we will discuss this comprehensively in the following subsection. It can be verified that the total mass of $\tilde{\mu}_k$ is conserved as the following equality holds:

$$\sum_{i=1}^M \eta \bar{U}_{\tilde{\mu}_{k+1/2}}(\mathbf{x}_{k+1}^i) a_k^i = 0,$$

Thus, $\tilde{\mu}_k$ remains a valid probability distribution during the whole procedure of DPVI, i.e. $\sum_i a_k^i = 1$ for all k.

In developing our DPVI framework, we adopt a Gauss-Siedel-type strategy to update the position $\mathbf{x}_k^i$ and the weight a_k^i, i.e., we adjust the weight based on the newly obtained position $\mathbf{x}_{k+1}^i$ in the k-th iteration. As the weight adjustment step (10) only involves calculating the first variation approximation $U_{\tilde{\mu}_{k+1/2}}$, it would bring *little* extra computational cost compared with the position update step (9), which involves calculating the gradient of $U_{\tilde{\mu}_k}$. It can be verified that, the overall computational complexity of DPVI is in the same order as their fixed weight counterpart, as the cost of gradient calculation in (9) is typically d times of the cost needed to compute function values in (10). Besides, one can further reduce the computational cost by adopting the Jacobi-type strategy, i.e., update weight a_k^i in (10) with $U_{\tilde{\mu}_k}$ at position $\mathbf{x}_k^i$. In this case, the weight adjustment step boils down to M scalar additions since the term $U_{\tilde{\mu}_k}$ can be directly obtained when calculating $\mathbf{v}_{\tilde{\mu}_k} = \nabla U_{\tilde{\mu}_k}$ in the position update step (9). The empirical result in the experiment section also shows that the extra weight-adjustment step would not bring much extra computational cost.

3.3 DPVI Algorithms and Their Duplicate/Kill Variants

To derive an efficient DPVI algorithm, it remains to decide the underlying functional $\mathcal{F}$, and construct proper empirical approximations to its first variation U_μ and the associated vector field $\mathbf{v}_\mu = -\nabla U_\mu$ (denoted as $U_{\tilde{\mu}}$ and $\mathbf{v}_{\tilde{\mu}}$, respectively). Unfortunately, there exists no systematic approach to design proper approximations $U_{\tilde{\mu}}$ and $\mathbf{v}_{\tilde{\mu}}$ for arbitrary dissimilarity functional $\mathcal{F}$. Here, we propose three efficient DPVI algorithms based on different approximations utilized in existing fixed-weight ParVIs, two with the KL-divergence as the underlying functional $\mathcal{F}$ (D-GFSD-CA and D-Blob-CA) and one with the KSD (D-KSDD-CA). Note that these three algorithms are actually the dynamic weight-adjustment counterparts of the fixed weight ParVI algorithms, GFSD, Blob and KSDD, respectively. Due to the limit of space, we only list the D-GFSD-CA algorithm here, and refer the readers to our long version for the details of the other two algorithms. Moreover, we emphasize that other divergences, such as MMD and W_2 (which are more related with the generative task), can also be chosen as the underlying functional.

Algorithm 1 Dynamic-weight Particle-based Variational Inference (DPVI) Framework

Input: Initial distribution $\tilde{\mu}_0 = \sum_{i=1}^M a_0^i \delta_{\mathbf{x}_0^i}$, step-size η, weight parameter λ.

1: **for** $k = 0, 1, ..., T-1$ **do**
2: **for** $i = 1, 2, ..., M$ **do**
3: Update positions $\mathbf{x}_{k+1}^i$'s according to (9) .
4: **end for**
5: **for** $i = 1, 2, ..., M$ **do**
6: Adjust weights a_{k+1}^i's according to (10).
7: **end for**
8: **end for**
9: **Output**:$\tilde{\mu}_T = \sum_{i=1}^M a_T^i \delta_{\mathbf{x}_T^i}$.

D-GFSD-CA algorithm. As we have discussed in the preliminaries, a large portion of existing fixed-weight ParVIs adopt the KL-divergence as underlying functional $\mathcal{F}$, whose associated vector field and first variation are defined as follows:

$$\mathbf{v}_\mu(\mathbf{x}) = \nabla \log \pi(\mathbf{x}) - \nabla \log \mu(\mathbf{x}),$$
$$U_\mu(\mathbf{x}) = \log \mu(\mathbf{x}) - \log \pi(\mathbf{x}).$$

In order to deal with the intractable $\log \mu(\mathbf{x})$ which is undefined with empirical distribution $\tilde{\mu}_k$, Our proposed D-GFSD-CA algorithm adopts the approximation techniques used in the fixed-weight ParVI methods GFSD Specifically, we directly approximate μ by smoothing the empirical distribution $\tilde{\mu}$ with a kernel function K: $\hat{\mu} = \tilde{\mu} * K = \sum_{i=1}^M a^i K(\cdot, \mathbf{x}^i)$, which leads to the following approximations:

$$\mathbf{v}_{\tilde{\mu}_k}(\mathbf{x}) = \nabla \log \pi(\mathbf{x}) - \frac{\sum_{i=1}^M a_k^i \nabla_{\mathbf{x}} K(\mathbf{x}, \mathbf{x}_k^i)}{\sum_{i=1}^M a_k^i K(\mathbf{x}, \mathbf{x}_k^i)}, \tag{11}$$

$$U_{\tilde{\mu}_{k+1/2}}(\mathbf{x}) = -\log \pi(\mathbf{x}) + \log \sum_{i=1}^M a_k^i K(\mathbf{x}, \mathbf{x}_{k+1}^i). \tag{12}$$

The duplicate/kill variants. In fact, there is a probabilistic discretization strategy of the approximate reaction flow in (7). This strategy duplicates/kills particle $\mathbf{x}_{k+1}^i$ according to an exponential clock with instantaneous rate:

$$R_{k+1}^i = -\lambda \eta \bar{U}_{\tilde{\mu}_{k+1/2}}(\mathbf{x}_{k+1}^i). \tag{13}$$

Specifically, if $R_{k+1}^i > 0$, duplicate the particle $\mathbf{x}_{k+1}^i$ with probability $1 - \exp(-R_{k+1}^i)$, and kill another one with uniform probability to conserve the total mass; if $R_{k+1}^i < 0$, kill the particle $\mathbf{x}_{k+1}^i$ with probability $1 - \exp(R_{k+1}^i)$, and duplicate another one with uniform probability. By replacing the CA strategy (10) in the DPVI framework, we could obtain the DK variants of DPVIs.

Algorithm	Number of particles				
	32	64	128	256	512
ULD	3.507	3.296	3.159	2.996	2.858
BDLS	3.653	3.185	2.943	2.678	2.581
HMC	3.504	3.181	3.025	2.744	2.540
SVGD	3.247	3.024	3.034	2.883	2.710
GFSD	3.364	3.315	3.335	3.198	3.050
D-GFSD-CA	2.855	2.639	2.510	2.388	2.272
D-GFSD-DK	3.006	2.744	2.593	2.422	2.304
Blob	3.249	3.207	3.195	3.047	2.884
D-Blob-CA	**2.651**	**2.493**	**2.356**	2.216	2.085
D-Blob-DK	2.716	2.501	2.346	**2.204**	**2.071**
KSDD	3.657	3.576	3.354	3.041	2.863
D-KSDD-CA	3.627	3.559	3.287	2.946	2.803
D-KSDD-DK	3.580	3.561	3.280	2.873	2.750

Table 1: Averaged W_2 distances with different number of particles.

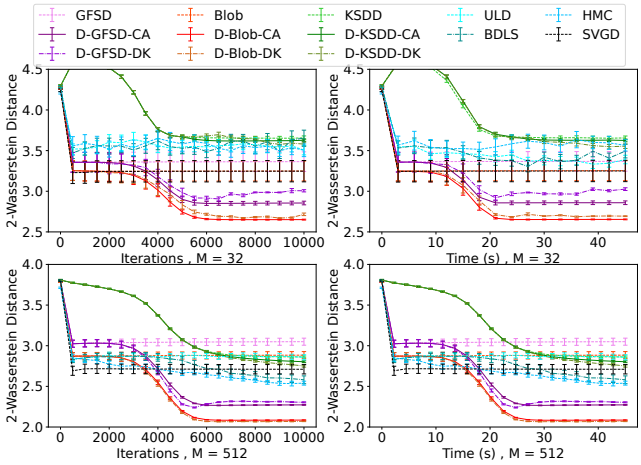

Figure 2: W_2 distance to the target w.r.t. iterations and time.

4 Experiments

In this section, we conduct empirical studies with our DPVI algorithms (D-GFSD-CA, D-Blob-CA, and D-KSDD-CA), their duplicate/kill variants (D-GFSD-DK, D-Blob-DK, and D-KSDD-DK), and the fixed-weight ParVI algorithms (SVGD, GFSD, Blob and KSDD). Besides, we also include three MCMC algorithms as our baseline: the Hamiltonian Monte Carlo (HMC) method [Neal and others, 2011], the Unadjusted Langevin Dynamics (ULD) method [Ma *et al.*, 2015], and the Birth Death Langevin Sampling (BDLS) method [Lu *et al.*, 2019] (ULD with DK weight-adjustment). We compare the performance of these algorithms on two simulations, i.e., a 10-D Single-mode Gaussian model (SG) and a Gaussian mixture model (GMM), and two real-world applications, i.e. a Gaussian Process (GP) regression and a Bayesian neural network. Due to limited space, only part of the results are reported in this section. We refer readers to our long version for the results on SG and additional results for GMM, GP and BNN.

4.1 Gaussian Mixture Model

We consider approximating a 10-D Gaussian mixture model with two components, weighted by $1/3$ and $2/3$ respectively. To investigate the influence of particle number M in fixed-weight ParVIs and the dynamic-weight algorithms, we run all the algorithms with $M \in \{32, 64, 128, 256, 512\}$.

In Table 1, we report the 2-Wasserstein (W_2) distance between the empirical distribution generated by each algorithm and the target distribution. We generate 5,000 samples from the target distribution π as reference to evaluate the W_2 distance by using the POT library [1]. It can be observed that both the CA and DK weight strategies contribute to obtain a higher approximation accuracy. The DPVI algorithms constantly outperform their fixed-weight counterparts with the same or even less number of particles. For instance, D-GFSD-CA with $M = 32$ achieves a lower W_2 than the fixed-weight GFSD algorithm with $M = 512$. Note that the KSDD-type algorithms (KSDD, D-KSDD-CA, and D-KSDD-DK) perform poorly in this task. Actually, this phenomenon has already been reported in KSDD [Korba *et al.*, 2021], and the authors claim that particles in KSDD may get stuck in spurious local minima when dealing with multi-mode models.

In Figure 2, we plot the W_2 w.r.t iterations and time of each algorithm. Due to the limit of space, we only report the results when $M = 32$ (the smallest) and $M = 512$ (the largest). From this figure, we can observe that, compared with the fixed-weight ParVI algorithms, both the CA and DK strategies result in a better performance w.r.t. both iterations and time. Actually, as we have discussed in the end of Section 3.2, while the weight-adjustment step in DPVI greatly enhances the expressiveness of particles' empirical distribution, it would not bring much extra computational costs compared with fixed-weight ParVI methods. Note that the performance of certain dynamic weight algorithms with DK (e.g., BDLS, D-GFSD-DK, and D-Blob-DK in the first subfigure) oscillates around its best value, which results from the fact that the step-size are tuned via grid search for the fixed-weight ParVI algorithm and then fixed for their dynamic-weight counterpart to show a direct influence of the weight adjustment strategy.

4.2 Gaussian Process Regression

The Gaussian Process (GP) model is widely adopted for the uncertainty quantification in regression problems [Rasmussen, 2003]. We follow the experiment setting in [Chen *et al.*, 2018b], and use the dataset LIDAR , which consists of 221 observations. In this task, we set the particle number to $M = 128$ for all the algorithms.

Figure 3 gives the contour line of the log posterior, and particles generated by each algorithm. It is shown that DPVIs with CA achieve better approximation results compared to other algorithms. We can also observe that D-Blob-CA has the best performance and covers a wider range of area due to both the dynamic weight adjustment strategy and extra repulsive term.

To evaluate how well the particles approximate the posterior $p(\phi|\mathcal{D})$, we also report the W_2 distance between the empirical distribution and the target distribution w.r.t iterations and time in Figure 4, given 10000 reference par-

[1] http://jmlr.org/papers/v22/20-451.html

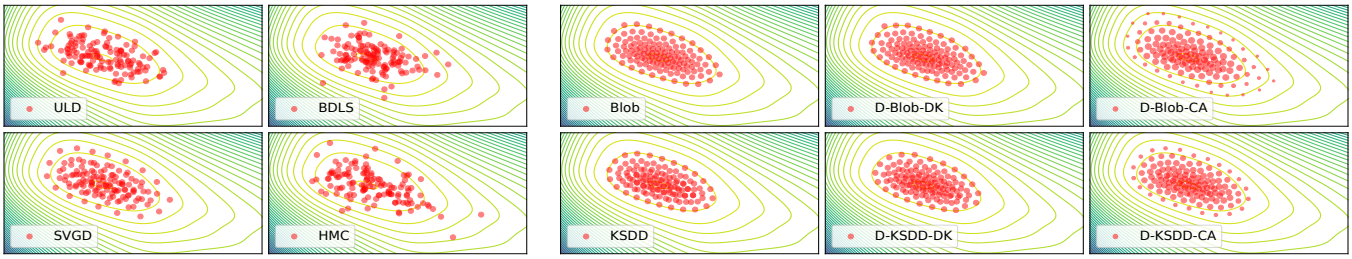

Figure 3: Approximation results in GP with 128 particles.

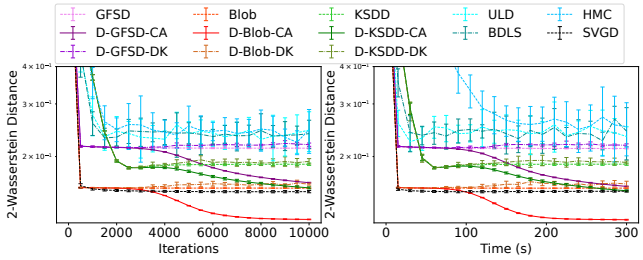

Figure 4: Results on approximating a Gaussian Process

Algorithm	Datasets			
	Electrical	Concrete	Kin8nm	WineRed
ULD	7.650E-3	6.254E+0	7.845E-2	6.430E-1
BDLS	7.532E-3	6.225E+0	7.826E-2	6.417E-1
SVGD	8.021E-3	6.119E+0	8.020E-2	6.338E-1
GFSD	7.572E-3	6.108E+0	7.870E-2	6.320E-1
D-GFSD-CA	**7.427E-3**	6.091E+0	**7.815E-2**	6.301E-1
D-GFSD-DK	7.477E-3	6.093E+0	7.826E-2	**6.291E-1**
Blob	7.572E-3	6.109E+0	7.872E-2	6.320E-1
D-Blob-CA	**7.427E-3**	**6.086E+0**	**7.815E-2**	6.301E-1
D-Blob-DK	7.491E-3	6.087E+0	7.829E-2	6.292E-1

Table 2: Averaged Test RMSE for the BNN task.

ticles generated by the HMC method after it achieves its equilibrium[Brooks *et al.*, 2011]. The result demonstrates the effectiveness of our proposed weight strategy: DPVIs with CA constantly outperform their fixed-weight counterparts, and D-Blob-CA achieves the highest approximation accuracy w.r.t. both iterations and time among all the algorithms. Note that the oscillation phenomena of algorithms with DK is more obvious than that in GMM . This is mainly due to the fact that the single-mode nature of GP greatly weakens the advantage of DK, i.e., transferring particles from low-probability region to distant high-probability area (e.g. among different local modes), and the gain of DK is not enough to overweight the influence of a sub-optimal stepsize.

4.3 Bayesian Neural Network

In this experiment, we study a Bayesian regression task with Bayesian neural network on 4 datasets from UCI and LIBSVM. We follow the experiment setting from [Liu and Wang, 2016], which models the output as a Gaussian distribution and uses a Gamma$(1, 0.1)$ prior for the inverse covariance. We use a one-hidden-layer neural network with 50 hidden units and maintain 128 particles. For all the datasets, we set the batchsize as 128. Since KSDD-type algorithms require evaluating the Hessian matrix of the objective function and the HMC method need calculate the full gradient of the log-posterior, which will induce enormous computational burden in neural network based tasks, we exclude them in this task.

We report the Root Mean Squared Error (RMSE) of each algorithm in Table 2, and we refer readers to Appendix for the negative log-likelihood results . The results show that both the CA and DK weight strategies contribute to a lower RMSE, and DPVI algorithms with CA achieve the best performance. It can be observed that GFSD-type algorithms obtain similar results as Blob-type algorithms, which may be ascribed to the negative correlation between the magnitude of the repulsive force and the dimensionality in ParVIs [Zhuo *et al.*, 2018].

5 Conclusion

In this paper, we propose a general Dynamic-weight Particle-based Variational Inference (DPVI) framework, which maintains a set of weighted particles to approximate a given target distribution π and updates both the particles' positions and weights iteratively. Our DPVI framework is developed by discretizing a novel composite flow, which is a combination of a finite-particle approximation of a reaction flow and the finite-particle position transport approximation adopted in existing fixed-weight ParVIs. We show that the mean-field limit of the proposed composite flow is actually the gradient flow of the associated dissimilarity functional $\mathcal{F}$ in the Wasserstein-Fisher-Rao space, which leads to an extra decrease of $\mathcal{F}$ than the Wasserstein gradient flow underlying existing fixed-weight ParVIs. We provide three effective DPVI algorithms with different finite-particle approximations, and derive three variants of them by using the duplicate/kill strategy. The empirical results show that the proposed DPVI algorithms constantly outperform their fixed-weight counterparts and D-Blob-CA usually obtains the best performance.

Acknowledgements

This work is supported by National Key Research and Development Program of China under Grant 2020AAA0107400, Zhejiang Provincial Natural Science Foundation of China (Grant No: LZ18F020002), Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, and National Natural Science Foundation of China (Grant No: 61672376, 61751209, 61472347).

References

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

[Brooks *et al.*, 2011] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

[Butcher, 1964] John C Butcher. Implicit runge-kutta processes. *Mathematics of Computation*, 18(85):50–64, 1964.

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

[Chen *et al.*, 2018a] Changyou Chen, Ruiyi Zhang, Wenlin Wang, Bai Li, and Liqun Chen. A unified particle-optimization framework for scalable bayesian sampling. *arXiv preprint arXiv:1805.11659*, 2018.

[Chen *et al.*, 2018b] Wilson Ye Chen, Lester Mackey, Jackson Gorham, François-Xavier Briol, and Chris Oates. Stein points. In *ICML*, pages 844–853. PMLR, 2018.

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

[Gallouët and Monsaingeon, 2017] Thomas O Gallouët and Leonard Monsaingeon. A jko splitting scheme for kantorovich–fisher–rao gradient flows. *SIAM Journal on Mathematical Analysis*, 49(2):1100–1130, 2017.

[Jordan *et al.*, 1998] 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.

[Kakutani, 1948] Shizuo Kakutani. On equivalence of infinite product measures. *Annals of Mathematics*, pages 214–224, 1948.

[Kondratyev *et al.*, 2016] Stanislav Kondratyev, Léonard Monsaingeon, Dmitry Vorotnikov, et al. A new optimal transport distance on the space of finite radon measures. *Advances in Differential Equations*, 21(11/12):1117–1164, 2016.

[Korba *et al.*, 2020] Anna Korba, Adil Salim, Michael Arbel, Giulia Luise, and Arthur Gretton. A non-asymptotic analysis for stein variational gradient descent. *NeurIPS*, 33, 2020.

[Korba *et al.*, 2021] Anna Korba, Pierre-Cyril Aubin-Frankowski, Szymon Majewski, and Pierre Ablin. Kernel stein discrepancy descent. *arXiv preprint arXiv:2105.09994*, 2021.

[Liero *et al.*, 2016] Matthias Liero, Alexander Mielke, and Giuseppe Savaré. Optimal transport in competition with reaction: The hellinger–kantorovich distance and geodesic curves. *SIAM Journal on Mathematical Analysis*, 48(4):2869–2911, 2016.

[Liu and Wang, 2016] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *arXiv preprint arXiv:1608.04471*, 2016.

[Liu and Wang, 2018] Qiang Liu and Dilin Wang. Stein variational gradient descent as moment matching. *arXiv preprint arXiv:1810.11693*, 2018.

[Liu and Zhu, 2018] Chang Liu and Jun Zhu. Riemannian stein variational gradient descent for bayesian inference. In *AAAI*, volume 32, 2018.

[Liu *et al.*, 2016] Qiang Liu, Jason Lee, and Michael Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In *ICML*, pages 276–284. PMLR, 2016.

[Liu *et al.*, 2019a] Chang Liu, Jingwei Zhuo, Pengyu Cheng, Ruiyi Zhang, and Jun Zhu. Understanding and accelerating particle-based variational inference. In *ICML*, pages 4082–4092, 2019.

[Liu *et al.*, 2019b] Chang Liu, Jingwei Zhuo, and Jun Zhu. Understanding mcmc dynamics as flows on the wasserstein space. In *ICML*, 2019.

[Liu, 2017] Qiang Liu. Stein variational gradient descent as gradient flow. *arXiv preprint arXiv:1704.07520*, 2017.

[Lu *et al.*, 2019] Yulong Lu, Jianfeng Lu, and James Nolen. Accelerating langevin sampling with birth-death. *arXiv preprint arXiv:1905.09863*, 2019.

[Ma *et al.*, 2015] Yi-An Ma, Tianqi Chen, and Emily B Fox. A complete recipe for stochastic gradient mcmc. *arXiv preprint arXiv:1506.04696*, 2015.

[Mroueh and Rigotti, 2020] Youssef Mroueh and Mattia Rigotti. Unbalanced sobolev descent. *NeurIPS*, 33, 2020.

[Neal and others, 2011] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

[Platen and Bruti-Liberati, 2010] Eckhard Platen and Nicola Bruti-Liberati. *Numerical solution of stochastic differential equations with jumps in finance*, volume 64. Springer Science & Business Media, 2010.

[Pu *et al.*, 2017] Yunchen Pu, Zhe Gan, Ricardo Henao, Chunyuan Li, Shaobo Han, and Lawrence Carin. Vae learning via stein variational gradient descent. In *NIPS*, 2017.

[Ranganath *et al.*, 2014] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.

[Rao, 1945] CR Rao. Information and accuracy attainable in the estimation of statistical parameters. kotz s & johnson nl (eds.), breakthroughs in statistics volume i: Foundations and basic theory, 235–248, 1945.

[Rasmussen, 2003] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

[Rotskoff *et al.*, 2019] Grant Rotskoff, Samy Jelassi, Joan Bruna, and Eric Vanden-Eijnden. Global convergence of neuron birth-death dynamics. *arXiv preprint arXiv:1902.01843*, 2019.

[Süli and Mayers, 2003] Endre Süli and David F Mayers. *An introduction to numerical analysis*. Cambridge university press, 2003.

[Wang and Li, 2019] Yifei Wang and Wuchen Li. Accelerated information gradient flow. *arXiv preprint arXiv:1909.02102*, 2019.

[Wibisono, 2018] Andre Wibisono. Sampling as optimization in the space of measures: The langevin dynamics as a composite optimization problem. In *COLT*, pages 2093–3027. PMLR, 2018.

[Zhang *et al.*, 2020a] Jianyi Zhang, Ruiyi Zhang, Lawrence Carin, and Changyou Chen. Stochastic particle-optimization sampling and the non-asymptotic convergence theory. In *Artificial Intelligence and Statistics*, pages 1877–1887. PMLR, 2020.

[Zhang *et al.*, 2020b] Jianyi Zhang, Yang Zhao, and Changyou Chen. Variance reduction in stochastic particle-optimization sampling. In *ICML*, pages 11307–11316. PMLR, 2020.

[Zhu *et al.*, 2020] Michael Zhu, Chang Liu, and Jun Zhu. Variance reduction and quasi-newton for particle-based variational inference. In *ICML*, pages 11576–11587. PMLR, 2020.

[Zhuo *et al.*, 2018] Jingwei Zhuo, Chang Liu, Jiaxin Shi, Jun Zhu, Ning Chen, and Bo Zhang. Message passing stein variational gradient descent. In *ICML*, pages 6018–6027. PMLR, 2018.