Particle-based Variational Inference with Preconditioned Functional Gradient Flow

Hanze Dong[1,*] **Xi Wang**[3] **Yong Lin**[2] **Tong Zhang**[1,2]
[1]Department of Mathematics, HKUST
[2]Department of Computer Science and Engineering, HKUST
[3]College of Information and Computer Science, UMass Amherst
[*]hdongaj@ust.hk

Abstract

Particle-based variational inference (VI) minimizes the KL divergence between model samples and the target posterior with gradient flow estimates. With the popularity of Stein variational gradient descent (SVGD), the focus of particle-based VI algorithms have been on the properties of functions in Reproducing Kernel Hilbert Space (RKHS) to approximate the gradient flow. However, the requirement of RKHS restricts the function class and algorithmic flexibility. This paper remedies the problem by proposing a general framework to obtain tractable functional gradient flow estimates. The functional gradient flow in our framework can be defined by a general functional regularization term that includes the RKHS norm as a special case. We also use our framework to propose a new particle-based VI algorithm: *preconditioned functional gradient flow* (PFG). Compared with SVGD, the proposed preconditioned functional gradient method has several advantages: larger function classes; greater scalability in the large particle-size scenarios; better adaptation to ill-conditioned target distribution; provable continuous-time convergence in KL divergence. Both theory and experiments have shown the effectiveness of our framework.

Remark: This is an extended abstract that summarizes the main results of the proposed framework and the conclusion. Full version is available at `https://arxiv.org/abs/2211.13954`.

1 Introduction

Given a target distribution $p_*(x)$, particle-based VI aims to find $g(t,x)$, so that starting with $X_0 \sim p_0$, the distribution $p(t,x)$ of the following method: $dX_t = g(t,X_t)dt$, converges to $p_*(x)$ as $t \to \infty$. By continuity equation [7], we can capture the evolution of $p(t,x)$ by

$$\frac{\partial p(t,x)}{\partial t} = -\nabla \cdot (p(t,x)g(t,x))\,. \tag{1}$$

In order to measure the "closeness" between $p(t,\cdot)$ and p_*, we typically adopt the KL divergence,

$$D_{\mathrm{KL}}(t) = \int p(t,x) \ln \frac{p(t,x)}{p_*(x)} dx. \tag{2}$$

Using chain rule and integration by parts, we have

$$\frac{dD_{\mathrm{KL}}(t)}{dt} = -\int p(t,x)[\nabla \cdot g(t,x) + g(t,x)^\top \nabla_x \ln p_*(x)]dx, \tag{3}$$

NeurIPS 2022 Workshop on Score-Based Methods.

which captures the evolution of KL divergence.

To minimize the KL divergence, one needs to define a "gradient" to update the particle distribution as our $g(t, x)$. The most standard approach, *Wasserstein gradient* [1], defines a gradient for $p(t, x)$ in the Wasserstein space, which contains probability measures with bounded second moment. In particular, for any functional $\mathcal{L}$ that maps probability density $p(t, x)$ to a non-negative scalar, we say that the particle density $p(t, x)$ follows the Wasserstein gradient flow of $\mathcal{L}$ if $g(t, x)$ is the $L^2(\mathbb{R}^d)$-functional derivative of $\mathcal{L}$. For KL divergence, the corresponding derivative is $\nabla \ln \frac{p_*(x)}{p(t,x)}$. However, the computation of deterministic and time-inhomogeneous Wasserstein gradient is non-trivial. It is necessary to restrict the function class of $g(t, x)$ to obtain a tractable form.

Stein variational gradient descent (SVGD) provides a tractable form to update particles with the kernelized gradient flow [2, 8]. It updates particles by minimizing the KL divergence with a functional gradient measured in RKHS. By restricting the functional gradient with bounded RKHS norm, it has an explicit formulation: $g(t, x)$ can be obtained by minimizing Eq. (3). Nonetheless, there are still some limitations due to the restriction of RKHS: (1) the expressive power is limited because kernel method is known to suffer from curse of dimensionality [5]; (2) with n particles, the $O(n^2)$ computational overhead of kernel matrix is required. Further, we identify another crucial limitation of SVGD: the kernel design is highly non-trivial. Even in the simple Gaussian case, where particles start with $\mathcal{N}(0, I)$ and $p_* = \mathcal{N}(\mu_*, \Sigma_*)$, commonly used kernels such as linear and RBF kernel, have fundamental drawbacks in SVGD algorithm (Example 1).

Our motivation originates from functional gradient boosting [4, 9, 6]. For each $p(t, x)$, we find a proper function as $g(t, x)$ in the function class $\mathcal{F}$ to minimize Eq. (3). In this context, we design a regularizer for the functional gradient to approximate variants of "gradient" explicitly. We propose a family of regularization to penalize the functional gradient output in the particle distribution. For well-conditioned $-\nabla^2 \ln p_*$[1], we can approximate the Wasserstein gradient directly; For ill-conditioned $-\nabla^2 \ln p_*$, we can adapt our regularizer to approximate a preconditioned one. Our functional gradient is an approximation to the preconditioned Wasserstein gradient. Regarding the function space, we do not restrict the function in RKHS. Instead, we use neural networks as our function classes to obtain better approximation capacity. The extension to the function space of neural networks gives the algorithm a much better expressive capacity, which will be justified with our empirical results.

Contributions. We propose a particle-based VI framework with regularized functional gradient flow. We choose a special family of regularizers to approximate preconditioned Wasserstein gradient, which is more effective than SVGD: The capacity of non-linear function classes are more expressive; The functional gradient in our framework explicitly approximate the preconditioned Wasserstein gradient, which supports ill-conditioned cases and obtains provable convergence rate; Our proposed algorithm does not need the $O(n^2)$ kernel matrix, leading to computational efficiency when particle size is large. Both theoretical and empirical results show the effectiveness of our framework.

2 PFG: Preconditioned Functional Gradient Flow

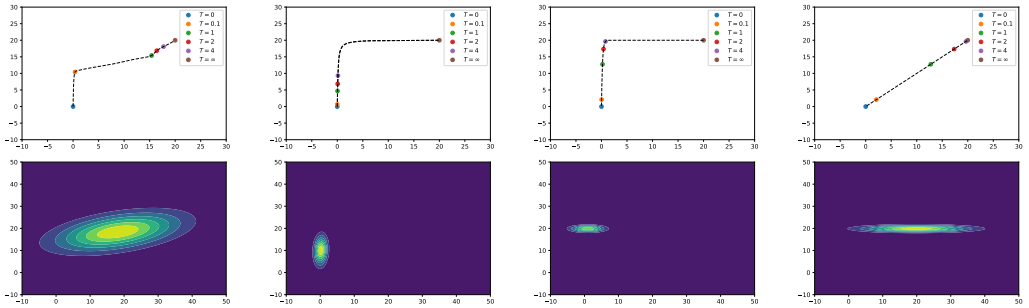
We let $g(t, x)$ belong to a vector-valued function class $\mathcal{F}$, and find the best gradient direction. Inspired by the gradient boosting algorithm for regression and classification problems, we approximate the Wasserstein gradient flow at any (t, x) by a function $g(t, x) \in \mathcal{F}$ which solves the following minimization formulation:

$$g(t, x) = \arg\min_{f \in \mathcal{F}} \left[- \int p(t, x)[\nabla \cdot f(x) + f(x)^\top \nabla \ln p_*(x)]dx + Q(f) \right], \tag{4}$$

where $Q(\cdot)$ is a regularization term that limit the output magnitude of f. This regularization term also implicitly determines the underlying "distance metric" used to define the gradient estimates $g(t, x)$ in our framework. When $Q(x) = \frac{1}{2}\|f\|^2_{\mathcal{H}}$ (RKHS norm), $g(t, x)$ is equivalent to kernelized gradient in SVGD. When $Q(x) = \frac{1}{2}\int p(t, x)\|f(x)\|^2 dx$, Eq. (4) is equivalent to

$$g(t, x) = \arg\min_{f \in \mathcal{F}} \int p(t, x) \left\| f(x) - \nabla \ln \frac{p_*(x)}{p(t, x)} \right\|^2 dx. \tag{5}$$

[1]For any matrix, condition number is the ratio of the maximal to the minimal eigenvalues. A low condition number is said to be well-conditioned, while a high condition number is said to be ill-conditioned.



(a) $Q(f) = \frac{1}{2}\|f\|^2_{\mathcal{H}^d}$ (linear) (b) $Q(f) = \frac{1}{2}\|f\|^2_{\mathcal{H}^d}$ (RBF) (c) $Q(f) = \frac{1}{2}\mathbf{E}_{p_t}\|f\|^2$ (d) $Q(f) = \frac{1}{2}\mathbf{E}_{p_t}\|f\|^2_{\Sigma_*^{-1}}$

Figure 1: Evolution of particle distribution from $\mathcal{N}([0,0]^\top, I)$ to $\mathcal{N}([20,20]^\top, \mathrm{diag}(100,1))$ (first row: evolution of particle mean μ_t; second row: particle distribution $p(5,x)$ at $t=5$)

Algorithm 1 PFG: Preconditioned Functional Gradient Flow

Input: Unnormalized target distribution $p_*(x) = e^{-U(x)}$, $f_\theta(x) : \mathbb{R}^d \to \mathbb{R}^d$, initial particles (parameters) $\{x_0^i\}_{i=1}^n$, θ_0, iteration parameter T, T', step size η, η', regularization function $h(\cdot)$.

for $t = 1, \ldots, T$ **do**

  Assign $\theta_t^0 = \theta_{t-1}$;

  for $t' = 1, \cdots, T'$ **do**

    Compute $\hat{L}(\theta) = \frac{1}{n}\sum_{i=1}^n \left(h(f_\theta(x_t^i)) + f_\theta(x_t^i)\cdot\nabla U(x_t^i) - \nabla\cdot f_\theta(x_t^i)\right)$

    Update $\theta_t^{t'} = \theta_t^{t'-1} - \eta'\nabla\hat{L}(\theta_t^{t'-1})$;

  end

  Assign $\theta_t = \theta_t^{T_1}$ and update particles $x_t^i = x_t^i + \eta\left(f_{\theta_t}(x_t^i)\right)$ for all $i = 1, \cdots, n$;

end

Return: Optimized particles $\{x_T^i\}_{i=1}^n$

If $\mathcal{F}$ is well-specified, i.e., $\nabla \ln \frac{p_*(x)}{p(t,x)} \in \mathcal{F}$, we have $g(t,x) = \nabla \ln \frac{p_*(x)}{p(t,x)}$, which is the direction of Wasserstein gradient. Interestingly, despite the computational intractability of Wasserstein gradient, Eq. (4) provides a tractable variational approximation.

Example 1. *Consider that $p(t,\cdot)$ is $\mathcal{N}(\mu_t, \Sigma_t)$, p_* is $\mathcal{N}(\mu_*, \Sigma_*)$. We consider the SVGD algorithm with linear kernel, RBF kernel, and regularized functional gradient formulation with $Q(f) = \frac{1}{2}\mathbf{E}_{p_t}\|f\|^2$, and $Q(f) = \frac{1}{2}\mathbf{E}_{p_t}\|f\|^2_{\Sigma_*^{-1}}$. Starting with $\mathcal{N}(0, I)$, the path of μ_t and $p(5,x)$ are illustrated in Fig. 1. The detailed mathematical derivation are provided in the full version.*

Example 1 shows the comparison of different regularizations. For RKHS norm, we consider the most common kernels: linear and RBF. Fig. 1 demonstrates the path of μ_t with different regularizers. For linear kernel, due to the curl component, $p(5,x)$ is rotated with an angle (Fig. 1 (a)). For RBF kernel, it is misspecified, leading to slow convergence, since linear function is not contained in the function class. The L_2 regularizer shows suboptimal performance due to the ill-conditioned Σ_*. We can see that $Q(f) = \frac{1}{2}\mathbf{E}_{p_t}\|f\|^2_{\Sigma_*^{-1}}$ produces the optimal path for μ_t (the line between μ_0 and μ_*).

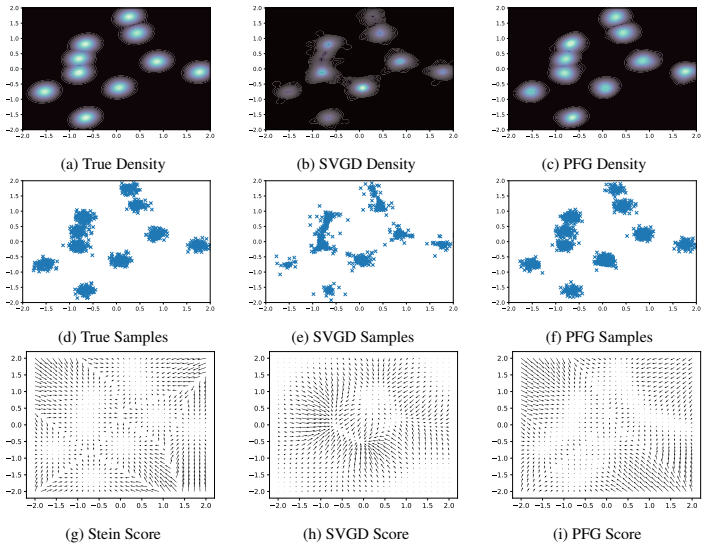
General Regularization. Inspired by the Gaussian case, we consider the general form

$$Q(f(x)) = \frac{1}{2}\int p(t,x)\|f(x)\|^2_H dx \tag{6}$$

where H is a symmetric positive definite matrix.

From the theoretical side, with sufficiently large function space, PFG (approximated preconditioned Wasserstein gradient flow) reaches linear convergence rate to approximate log-Sobolev distributions, which cannot be done by SVGD [3].

We will realize our algorithm with parametric f_θ and discretize the update. Full procedure is presented in Algorithm 1, where the regularizer h is $\frac{1}{2}\|\cdot\|^2_H$ by default.

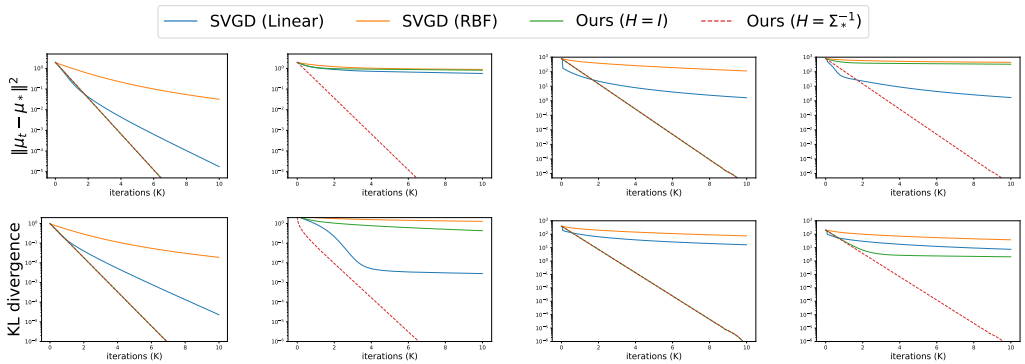


(a) True Density (b) SVGD Density (c) PFG Density

(d) True Samples (e) SVGD Samples (f) PFG Samples

(g) Stein Score (h) SVGD Score (i) PFG Score

Figure 2: Particle-based VI for Gaussian mixture sampling.



(a) $\mu_* = [1, 1]^\top$ $\Sigma_* = \mathrm{diag}(1, 1)$ (b) $\mu_* = [1, 1]^\top$ $\Sigma_* = \mathrm{diag}(100, 1)$ (c) $\mu_* = [20, 20]^\top$ $\Sigma_* = \mathrm{diag}(1, 1)$ (d) $\mu_* = [20, 20]^\top$ $\Sigma_* = \mathrm{diag}(100, 1)$

Figure 3: Evolution of particle distribution from $\mathcal{N}(0, I)$ to $\mathcal{N}(\mu_*, \Sigma_*)$ (first row: mean squared error of μ_t: $\|\mu_t - \mu_*\|^2$; second row: KL divergence between $p(t, x)$ and $p_*(x)$)

3 Experiments

Gaussian Mixture. To demonstrate the capacity of non-linear function class, we have conducted the Gaussian mixture experiments to show the advantage over linear function class, e.g., SVGD (RBF kernel). We consider to sample from a 10-cluster Gaussian Mixture distribution. Both SVGD and our algorithm are trained with 1,000 particles. Fig. 2 shows that the estimated score by RBF kernel is usually unsatisfactory: (1) In low-density area, it suffers from gradient vanishing, which makes samples stuck at these parts (b)); (2) The score function cannot distinguish connected clusters.

Ill-conditioned Gaussian distribution. We show the effectiveness of our proposed regularizer. For ill-conditioned case, the condition number (the ratio between maximal and minimal eigenvalue) of Σ_* is large. We compare different μ_* and Σ_*. When Σ_* is well-conditioned ($\Sigma_* = I$), L_2 regularizer performs well. However, it will be slowed down significantly with ill-conditioned Σ_*. For SVGD with linear kernel, the convergence slows down with shifted μ_* or ill-conditioned Σ_*. For SVGD with RBF kernel, the convergence is slow due to the misspecified function class. Interestingly, for ill-conditioned case, μ_t of SVGD (linear) converges faster than our method with $H = I$ but KL divergence does not always follow the trend. The reason is that Σ_t of SVGD is highly biased, making KL divergence large. Our algorithm ($H = \Sigma_*^{-1}$) extends the Wasserstein gradient and makes the particle-based sampling algorithm compatible with ill-conditioned sampling case.

References

[1] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005.

[2] Sinho Chewi, Thibaut Le Gouic, Chen Lu, Tyler Maunu, and Philippe Rigollet. Svgd as a kernelized wasserstein gradient flow of the chi-squared divergence. *Advances in Neural Information Processing Systems*, 33:2098–2109, 2020.

[3] Andrew Duncan, Nikolas Nüsken, and Lukasz Szpruch. On the geometry of stein variational gradient descent. *arXiv preprint arXiv:1912.00894*, 2019.

[4] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[5] Gery Geenens. Curse of dimensionality and related issues in nonparametric functional regression. *Statistics Surveys*, 5:30–43, 2011.

[6] Rie Johnson and Tong Zhang. A framework of composite functional gradient methods for generative adversarial models. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):17–32, 2019.

[7] 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.

[8] Qiang Liu. Stein variational gradient descent as gradient flow. *Advances in neural information processing systems*, 30, 2017.

[9] Atsushi Nitanda and Taiji Suzuki. Gradient layer: Enhancing the convergence of adversarial training for generative models. In *International Conference on Artificial Intelligence and Statistics*, pages 1008–1016. PMLR, 2018.

Acknowledgements

The work was supported by the General Research Fund (GRF 16310222 and GRF 16201320).