A Unified Particle-Optimization Framework for Scalable Bayesian Sampling

Changyou Chen
University at Buffalo (cchangyou@gmail.com)
Ruiyi Zhang, Wenlin Wang, Bai Li, Liqun Chen
Duke University

Abstract

There has been recent interest in developing scalable Bayesian sampling methods such as stochastic gradient MCMC (SG-MCMC) and Stein variational gradient descent (SVGD) for big-data analysis. A standard SG-MCMC algorithm simulates samples from a discrete-time Markov chain to approximate a target distribution, thus samples could be highly correlated, an undesired property for SG-MCMC. In contrast, SVGD directly optimizes a set of particles to approximate a target distribution, and thus is able to obtain good approximations with relatively much fewer samples. In this paper, we propose a principle particle-optimization framework based on Wasserstein gradient flows to unify SG-MCMC and SVGD, and to allow new algorithms to be developed. Our framework interprets SG-MCMC as particle optimization on the space of probability measures, revealing a strong connection between SG-MCMC and SVGD. The key component of our framework is several particle-approximate techniques to efficiently solve the original partial differential equations on the space of probability measures. Extensive experiments on both synthetic data and deep neural networks demonstrate the effectiveness and efficiency of our framework for scalable Bayesian sampling.

1 INTRODUCTION

Bayesian methods have been playing an important role in modern machine learning, especially in unsupervised learning (Kingma and Welling, 2014; Li et al., 2017), and recently in deep reinforcement learning (Houthooft et al., 2016; Liu et al., 2017). When dealing with big data, two lines of research directions have been developed to scale up Bayesian methods, e.g., variational-Bayes-based and sampling-based methods. Stochastic gradient Markov chain Monte Carlo (SG-MCMC) is a family of scalable Bayesian learning algorithms designed to efficiently sample from a target distribution such as a posterior distribution (Welling and Teh, 2011; Chen et al., 2014; Ding et al., 2014; Chen et al., 2015). In principle, SG-MCMC generates samples from a Markov chain, which are used to approximate a target distribution. Under a standard setting, samples from SG-MCMC are able to match a target distribution exactly with an infinite number of samples (Teh et al., 2016; Chen et al., 2015). However, this is practically infeasible, as only a finite number of samples are obtained. Although nonasymptotic approximation bounds w.r.t. the number of samples have been investigated (Teh et al., 2016; Vollmer et al., 2016; Chen et al., 2013), there are no theory/algorithms to guide learning an optimal set of fixed-size samples/particles. This is an undesirable property of SG-MCMC, because in practice one often seeks to learn the optimal samples of a finite size that best approximate a target distribution.

A remedy for this issue is to adopt the idea of particle-based sampling methods, where a set of particles (or samples) are initialized from some simple distribution, followed by iterative updates to better approximate a target distribution. The updating procedure is usually done by optimizing some metrics such as a distance measure between the target distribution and the current approximation. There is not much work in this direction for large-scale Bayesian sampling, with an outstanding representative being the Stein variational gradient descent (SVGD) (Liu and Wang, 2016). In SVGD, the update of particles are done by optimizing the KL-divergence between the empirical particle distribution and a target distribution, thus the samples are designed to be updated optimally to reduce the KL-divergence in each iteration. Because of this property, SVGD is found to perform better than SG-MCMC when the number of samples used to approximate a target distribution is limited, and has been
applied to other problems such as deep generative models (Feng et al., 2017) and deep reinforcement learning (Liu et al., 2017; Haarnoja et al., 2017; Zhang et al., 2018b).

Though often achieving comparable performance in practice, little work has been done on investigating connections between SG-MCMC and SVGD, and on developing particle-optimization schemes for SG-MCMC. In this paper, adopting ideas from Wasserstein-gradient-flow literature, we propose a unified particle-optimization framework for scalable Bayesian sampling. The idea of our framework is to work directly on the evolution of a density functions on the space of probability measures, e.g., the Fokker-Planck equation in SG-MCMC. To make the evolution solution computationally feasible, particle approximations are adopted for densities, where particles can be optimized during the evolution process. Both SG-MCMC and SVGD are special cases of our framework, and are shown to be highly related. Notably, sampling with SG-MCMC becomes a deterministic particle-optimization problem as SVGD on the space of probability measures, overcoming the aforementioned correlated-sample issue. Furthermore, we are able to develop new unified particle-optimization algorithms by combining SG-MCMC and SVGD, which is less prone to high-dimension space and thus obtains better performance for large-scale Bayesian sampling. We conduct extensive experiments on both synthetic data and Bayesian learning of deep neural networks, verifying the effectiveness and efficiency of our proposed framework.

2 PRELIMINARIES

In this section, we review related concepts and algorithms for SG-MCMC, SVGD, and Wasserstein gradient flows (WGF) on the space of probability measures.

2.1 Stochastic gradient MCMC

Diffusion-based sampling methods Generating random samples from a distribution (e.g., a posterior distribution) is one of the fundamental problems in Bayesian statistics, which has many important applications in machine learning. Traditional Markov Chain Monte Carlo methods (MCMC), such as the Metropolis–Hastings algorithm (Metropolis et al., 1953) produces unbiased samples from a desired distribution when the density function is known up to a normalizing constant. However, most of these methods are based on random walk proposals which suffer from high dimensionality and often lead to highly correlated samples. On the other hand, dynamics-based sampling methods such as the Metropolis adjusted Langevin algorithm (MALA) (Xifara et al., 2014) avoid this high degree of correlation by combining dynamical systems with the Metropolis step. In fact, these dynamical systems are derived from a more general mathematical technique called diffusion process, or more specifically, Itô diffusion (Øksendal [1985]).

Specifically, our objective is to generate random samples from a posterior distribution \( p(\theta | X) \propto p(X | \theta)p(\theta) \), where \( \theta \in \mathbb{R}^r \) represents the model parameter, and \( X \triangleq \{x_i\}_{i=1}^N \) represents the data. The canonical form is \( p(\theta | X) = (1/Z) \exp(U(\theta)) \), where \( U(\theta) = \log p(X | \theta) + \log p(\theta) \)

is referred to as the potential energy based on an i.i.d. assumption of the model, and \( Z \) is the normalizing constant. In Bayesian sampling, the posterior distribution corresponds to the marginal stationary distribution of a (continuous-time) Itô diffusion, defined as a stochastic differential equation of the form:

\[
\mathrm{d}\Theta_t = F(\Theta_t)\mathrm{d}t + g(\Theta_t)\mathrm{d}W_t, \tag{1}
\]

where \( t \) is the time index; \( \Theta_t \in \mathbb{R}^p \) represents the full variables in a dynamical system, and \( \Theta_t \geq \Theta_t \) (thus \( p \geq r \)) is potentially an augmentation of model parameter \( \theta \); \( W_t \in \mathbb{R}^p \) is a deterministic Brownian motion. Functions \( F: \mathbb{R}^p \rightarrow \mathbb{R}^p \) and \( g: \mathbb{R}^p \rightarrow \mathbb{R}^{p \times p} \) are assumed to satisfy the Lipschitz continuity condition (Ghosh, 2011). By Fokker-Planck equation (or the forward Kolmogorov equation) (Kolmogorov, [1931] [Risken, 1989]), when appropriately designing the diffusion-coefficient functions \( F(\cdot) \) and \( g(\cdot) \), the stationary distribution of the corresponding Itô diffusion equals the posterior distribution of interest, \( p(\theta | X) \). For example, the 1st-order Langevin dynamics defines \( \Theta_t = \theta, F(\Theta_t) = \frac{1}{2} \nabla_\theta U(\theta), \) \( g(\Theta_t) = I_r \); the 2nd-order Langevin diffusion defines \( \Theta_t = (\theta, q) \), and \( F(\Theta_t) = \left( \begin{array}{cc} -Bq & -\nabla_\theta U(\theta) \end{array} \right), g(\Theta_t) = \sqrt{2B} \left( \begin{array}{cc} 0 & 0 \\ 0 & I_r \end{array} \right) \) for a scalar \( B > 0 \); \( q \) is an auxiliary variable known as the momentum (Chen et al., 2014; Ding et al., 2014).

Let the density of \( \Theta_t \) be \( \mu_t \), it is known \( \mu_t \) is characterized by the Fokker-Planck (FP) equation \( \text{[Risken, 1989]} \):

\[
\frac{\partial \mu_t}{\partial t} = -\nabla_\Theta \cdot (\mu_t F(\Theta_t)) + \nabla_\Theta \nabla_\Theta : (\mu_t \Sigma(\Theta_t)) \tag{2}
\]

where \( \Sigma(\Theta_t) \triangleq g(\Theta_t)g^T(\Theta_t), A \cdot B \triangleq \text{trace}(A^T B) \) for matrices \( A \) and \( B \). The FP equation is the key to develop our particle-optimization framework for SG-MCMC. In the following, we focus on the simplest case of 1st-order Langevin dynamics if not stated explicitly, though the derivations apply to other variants.

Stochastic gradient MCMC SG-MCMC algorithms are discretized numerical approximations of the Itô dif-
fusion \cite{lee2012unified}. They mitigate the slow mixing and non-
scalability issues encountered in traditional MCMC algo-

\textit{ri}thms by \textit{i}) adopting gradient information of the posterior

distribution, \textit{ii}) using minibatches of the data in each iter-

ation of the algorithm to generate samples, and \textit{iii}) ignor-

ing the rejection step as in standard MCMC. To make the

algorithms scalable in a big-data setting, three develop-

ments will be implemented based on the Itô diffusion: \textit{i})
define appropriate functions \(F\) and \(g\) in the Itô-diffusion

formula so that the (marginal) stationary distributions co-

incide with the target posterior distribution \(p(\theta | X)\); \textit{ii})
replace \(F\) or \(g\) with unbiased stochastic approximations

to reduce the computational complexity, e.g., approximat-

ing \(F\) with a random subset of the data instead of using the

full data. For example, in the 1st-order Langevin dy-

namics, \(\nabla_\theta U(\theta)\) could be \textit{approximated} by an unbiased
estimator with a subset of data:

\[
\nabla_\theta \hat{U}(\theta) = \nabla \log p(\theta) + \frac{\nu}{n} \sum_{i=1}^{n} \nabla \log p(x_i | \theta) \tag{3}
\]

where \(\pi\) is a size-\(n\) random subset of \(\{1, 2, \cdots, N\}\), lead-
ing to the first SG-MCMC algorithm in machine learning –

stochastic gradient Langevin dynamics (SGLD) \cite{welling2011bayesian}; and \textit{iii}) solve the generally intractable

continuous-time Itô diffusions with a numerical method,
e.g., the Euler method \cite{chen2015stochastic}. For example,
this leads to the following update in SGLD:

\[
\theta_\ell = \theta_{\ell-1} + \nabla_\theta \hat{U}(\theta_{\ell-1}) h + \sqrt{2h} \delta_\ell,
\]

where \(h\) means the stepsize, \(\ell\) indexes the samples, \(\delta_\ell \sim \mathcal{N}(0, I)\) is a random sample from an isotropic normal

distribution. After running the algorithm for \(L\) steps, the

collection of samples \(\{\theta_\ell\}_{\ell=1}^{L}\) are used to approximate the unknown posterior distribution \(\frac{1}{Z} e^{U(\theta)}\).

\section{Stein variational gradient descent}

Different from SG-MCMC, SVGD initializes a set of par-
ticles which are iteratively updated so that the empirical

particle distribution approximates the posterior distribu-
tion. Specifically, we consider a set of particles \(\{\theta^{(i)}\}_{i=1}^{M}\)
drawn from some distribution \(q\). SVGD tries to update

these particles by doing gradient descent on the interactive

particle system via

\[
\theta^{(i)} \leftarrow \theta^{(i)} + h \phi(\theta^{(i)}), \quad \phi = \arg \max_{\phi \in \mathcal{F}} \frac{\partial}{\partial h} \text{KL}(q(h\phi || p))\]

where \(\phi\) is a function perturbation direction chosen to

minimize the KL divergence between the updated density

\(q(h\phi)\) estimated by the particles and the posterior \(p(\theta | X)\)

\(p\) for short). Since \(\text{KL}(q || p)\) is convex in \(q\), global opti-

mum of \(q = p\) can be guaranteed. SVGD considers \(\mathcal{F}\) as

the unit ball of a vector-valued reproducing kernel Hilbert

space (RKHS) \(\mathcal{H}\) associated with a kernel \(\kappa(\theta, \theta')\). In

such as setting \cite{liu2016stein} shown:

\[
\frac{\partial}{\partial h} \text{KL}(q(h\phi || p)) |_{h=0} = \mathbb{E}_{\theta \sim q} \left[ \text{trace}(\Gamma_p \phi(\theta)) \right], \tag{4}
\]

where \(\Gamma_p\) is the Stein operator. Assuming that the update

function \(\phi(\theta)\) is in a RKHS with kernel \(\kappa(\cdot, \cdot)\), it was

shown in \cite{liu2016stein} that (4) is maximized with:

\[
\phi(\theta) = \mathbb{E}_{\theta \sim q} [\kappa(\theta, \theta') \nabla_\theta U(\theta) + \nabla_\theta \kappa(\theta, \theta')]. \tag{5}
\]

When approximating the expectation \(\mathbb{E}_{\theta \sim q} [\cdot]\) with empiri-
cal particle distribution and adopting stochastic gradients,

we arrive at the following updates for the particles (\(\ell\)

donotes the iteration number): \(\theta^{(i)}_{\ell+1} = \theta^{(i)}_{\ell} +

\frac{h}{M} \sum_{j=1}^{M} \left[ \kappa(\theta^{(j)}_{\ell}, \theta^{(i)}_{\ell}) \nabla_\theta \hat{U}(\theta^{(j)}_{\ell}) + \nabla_\theta \kappa(\theta^{(j)}_{\ell}, \theta^{(i)}_{\ell}) \right] \tag{6}
\]

SVGD applies updates (6) repeatedly, moving the sam-

els to a target distribution \(p\).

\section{Wasserstein Gradient Flows}

For a better motivation of WGF, we start from gradient
flows defined on the Euclidean space.

\subsection{Gradient flows on the Euclidean space}

For a smooth \(E : \mathbb{R}^r \rightarrow \mathbb{R}\), and a starting point \(\theta_0 \in \mathbb{R}^r\), the gradient
flow of \(E(\theta)\) is defined as the solution of the differ-
ential equation: \(\frac{d\theta}{dt} = -\nabla E(\theta(t))\), s.t. \(\theta(0) = \theta_0\).

This is a standard Cauchy problem \cite{ruggieri1995solution}, en-
dowed with a unique solution if \(\nabla E\) is Lipschitz con-
tinuous. When \(E\) is non-differentiable, the gradient is

replaced with its subgradient, defined as \(\partial E(\theta) \equiv \{p \in
\mathbb{R}^r : F(\theta') \geq F(\theta) + p \cdot (\theta' - \theta), \forall \theta' \in \mathbb{R}^r\}\). Note

\(\partial E(\theta) = \{\nabla E(\theta)\}\) if \(E\) is differentiable at \(\theta\). In this

case, the gradient flow formula above is replaced with:

\(\frac{d\theta}{dt} \in -\partial E(\theta(t))\).

\subsection{Wasserstein gradient flows}

Let \(\mathcal{P}(\Omega)\) denote the space of probability measures on \(\Omega \subset \mathbb{R}^r\). WGF is an exten-
sion of gradient flows in Euclidean space by lifting the

definition onto the space of probability measures. Form-
ally, let \(\mathcal{P}(\Omega)\) be endowed with a Riemannian geometry

induced by the 2nd-order Wasserstein distance, \(\text{i.e.}, the\)
curve length between two elements (two distributions) is

defined as:

\[
W_2^2(\mu, \nu) \equiv \inf_{\gamma} \int_{\Omega \times \Omega} \|\theta - \theta'\|^2 d\gamma(\theta, \theta') : \gamma \in \Gamma(\mu, \nu)
\]

where \(\Gamma(\mu, \nu)\) is the set of joint distributions over \((\theta, \theta')\)
such that the two marginals equal \(\mu\) and \(\nu\), respectively.
The Wasserstein distance can be explained as an optimal-
transport problem, where one wants to transform elements
in the domain of $\mu$ to $\nu$ with a minimum cost [Villani 2008]. The term $\|\theta - \theta\|^2_2$ represents the cost to transport $\theta$ in $\mu$ to $\theta'$ in $\nu$, and can be replaced by a general metric $c(\theta, \theta')$ in a metric space. If $\mu$ is absolutely continuous w.r.t. the Lebesgue measure, there is a unique optimal transport plan from $\mu$ to $\nu$, i.e., a mapping $T : \mathbb{R}^r \rightarrow \mathbb{R}^r$ pushing elements in the domain of $\mu$ onto $\nu$ satisfying $T_\#\mu = \nu$. Here $T_\#\mu$ denotes the pushforward measure of $\mu$. The Wasserstein distance is equivalently reformulated as: $W^2(\mu, \nu) \triangleq \inf_{\nu \in \mathcal{P}(\mathbb{R}^r)} \left\{ \int c(\theta, T(\theta)) d\nu(\theta) \right\}$.

Let $\{\mu_t\}_{t \in [0,1]}$ be an absolutely-continuous curve in $\mathcal{P}(\Omega)$ with finite second-order moments. We consider to define the change of $\mu_t$’s by investigating $W^2(\mu_t, \mu_{t+h})$. Motivated by the Euclidean-space case, this is reflected by a vector field, $v_t(\theta) \triangleq \lim_{h \rightarrow 0} \frac{T_\#\mu_t(\theta) - T_\#\mu_{t+h}(\theta)}{h}$, called the velocity of the particle. A gradient flow can be defined on $\mathcal{P}(\Omega)$ correspondingly [Ambrosio et al. 2005].

**Lemma 1** Let $\{\mu_t\}_{t \in [0,1]}$ be an absolutely-continuous curve in $\mathcal{P}(\Omega)$ with finite second-order moments. Then for a.e. $t \in [0,1]$, the above vector field $v_t$ defines a gradient flow on $\mathcal{P}(\Omega)$ as: $\partial_t \mu_t + \nabla \cdot (v_t \mu_t) = 0$.

The gradient flow describes the evolution of a functional $E$, which is a lifted version of the function in the case of Euclidean space in Section 2.3 to the space of probability measures. $E$ maps a probability measure $\mu$ to a real value, i.e., $E : \mathcal{P}(\Omega) \rightarrow \mathbb{R}$. We will focus on the case where $E$ is convex in this paper, which is enough considering gradient flows for SG-MCMC and SVGD, though the theory applies to a more general $\lambda$-convex energy functional setting [Ambrosio et al. 2005]. It can be shown that $v_t$ in Lemma 1 has the form $v_t = -\nabla \frac{\delta E}{\delta \mu_t}(\mu_t)$ (Ambrosio et al. 2005), where $\delta E_{\mu_t}$ is called the first variation of $E$ at $\mu_t$. Based on this, gradient flows on $\mathcal{P}(\Omega)$ can be written as:

$$\partial_t \mu_t = -\nabla \cdot (v_t \mu_t) = \nabla \cdot \left( \mu_t \nabla \left( \frac{\delta E}{\delta \mu_t}(\mu_t) \right) \right).$$  \hspace{1cm} (7)

**Remark 1** Intuitively, an energy functional $E$ characterizes the landscape structure (appearance) of the corresponding manifold in $\mathcal{P}(\Omega)$, and the gradient flow (7) defines a geodesic path on this manifold. Usually, by choosing appropriate $E$, the landscape is convex, e.g., for the cases of both SG-MCMC and SVGD described below. This provides a theoretical guarantee on the optimal convergence of a gradient flow.

### 3 PARTICLE-OPTIMIZATION-BASED SAMPLING

In this section, we interpret the continuous versions of both SG-MCMC and SVGD as WGFs, followed by several techniques for particle optimization in the next section. In the following, $\mu_t$ denotes the distribution of $\theta_t$.

### 3.1 SVGD as WGF

The continuous-time and infinite-particle limit of SVGD with full gradients, denoted as SVGD$^\infty$, is known to be a special instance of the Vlasov equation in nonlinear partial-differential-equation literature [Liu 2017]:

$$\partial_t \mu_t = \nabla \cdot ((W * \mu_t)(\mu_t)),$$  \hspace{1cm} (8)

where $(W * \mu_t)(\theta) \triangleq \int W(\theta - \theta') \mu_t(\theta') d\theta'$ is the convolutional operator applied for some function $W : \mathbb{R}^r \rightarrow \mathbb{R}$. To specify SVGD$^\infty$, we generalize the convolutional operator, and consider $W$ as a function with two input arguments, i.e.,

$$W(\theta, \theta') \triangleq \int W(\theta - \theta') \mu_t(\theta') d\theta'.$$

Under this setting, we can specify the function $W(\cdot, \cdot)$ for SVGD$^\infty$ as

$$W(\theta, \theta') \triangleq \frac{\nabla \log p(\theta | X) \kappa(\theta', \theta) + \nabla \phi \kappa(\theta', \theta)}{\phi(\theta')}.\hspace{1cm} (9)$$

As will be shown in Section 4, this interpretation allows one to derive SVGD, a particle-optimization-based algorithm to approximate the continuous-time equation (8).

**Proposition 2** The stationary distribution of $W$ in (8) naturally leads to the SVGD algorithm, without the need to derive from an RKHS perspective.

To interpret SVGD$^\infty$ as a WGF, we need to specify two quantities, the energy functional and an underlying metric to measure distances between density functions.

**Energy functional and distance metric of SVGD$^\infty$**

There are two ways to derive energy functionals for SVGD$^\infty$, depending on the underlying metrics for probability distributions. When adopting the WGF framework where $W_2$ is used as the underlying metric, according to (7), the energy functional $E_\phi$ must satisfy

$$\nabla_\theta \left( \frac{\delta E_\phi}{\delta \mu_t}(\mu_t) \right) = W(\theta, \theta') * \mu_t,$$

$$\quad = \mathbb{E}_{\theta' \sim \mu_t} \left[ \nabla_{\theta'} \frac{p(\theta | X) \kappa(\theta, \theta')}{p(\theta')} \right] / p(\theta') \hspace{1cm} (10)$$

In general, there is no close-form solution for the above equation. Alternatively, Liu (2017) proved another form of the energy functional by defining a different distance metric on the space of probability measures, called $H$-Wasserstein distance:

$$W_H(q_1, q_2) \triangleq \inf_{\phi_t, \mu_t} \left\{ \int_0^1 \left\| \phi_t \|_H dt, \text{ s.t. } \mu_t = -\nabla_\theta \cdot (\phi_t \mu_t), \mu_0 = q_1, \mu_1 = q_2 \right\} \right\},$$

$$\quad \hspace{1cm} (11)$$

where $\phi_t \triangleq W * \mu_t$, and $\| \cdot \|_H$ is the norm in the Hilbert space induced by $\kappa(\cdot, \cdot)$. Under this metric, the underlying energy functional is proved to be the standard KL-divergence between $\mu_t$ and $\rho$, e.g.,

$$E_\rho = KL(\mu_t, p(\cdot | X)).$$

As can be seen in Section 4, this interpretation allows one to derive SVGD, a particle-optimization-based algorithm to approximate the continuous-time equation (8).
3.2 SG-MCMC as WGF

The continuous-time limit of SG-MCMC, when considering gradients to be exact, corresponds to standard Itô diffusions. We consider the Itô diffusion of SGLD for simplicity, e.g.,

$$d\theta_t = \frac{1}{2} \nabla U(\theta_t) dt + dW.$$  \hfill (12)

**Energy functional** The energy functional for SG-MCMC is easily seen by noting that the corresponding FP equation is in the gradient-flow form of (7). Specifically, the energy functional $E$ is defined as:

$$E(\mu) \triangleq -\int_{E_1} U(\theta) \mu(\theta) d\theta + \int_{E_2} \mu(\theta) \log \mu(\theta) d\theta$$  \hfill (13)

Note $E_2$ is the energy functional of a pure Brownian motion (e.g., $U(\theta) = 0$ in (12)). We can verify (13) by showing that it satisfies that FP equation. According to (7), the first variation of $E_1$ and $E_2$ is calculated as

$$\frac{\delta E_1}{\delta \mu} = -U, \quad \frac{\delta E_2}{\delta \mu} = \log \mu + 1. $$  \hfill (14)

Substituting (14) into (7) recovers the FP equation (2) for the Itô diffusion (12).

4 PARTICLE OPTIMIZATION

An efficient way to solve the generally infeasible WGF formula (7) is to adopt numerical methods with particle approximation. With a little abuse of notation but for conciseness, we do not distinguish subscripts $t$ and $\ell$ for the particle $\theta$, i.e., $\theta_\ell$ denotes the continuous-time version of the particle, while $\theta_t$ denotes the discrete-time version. We develop several techniques to approximate different types of WGF for SG-MCMC and SVGD. In particle approximation, the continuous density $\mu_t$ is approximated by a set of $M$ particles $(\theta^{(i)}_t)_{i=1}^M$ that evolve over time $t$ with weights $(m_i)_i$ such that $\sum_{i=1}^M m_i = 1$, i.e., $\mu_t \approx \sum_{i=1}^M m_i \delta(\theta^{(i)}_t)$, where $\delta(\theta^{(i)}_t) = 1$ when $\theta = \theta^{(i)}_t$ and 0 otherwise. Typically $m_i$'s are chosen at the beginning and fixed over time, thus we assume $m_i = \frac{1}{M}$ and rewrite $\mu_t \approx \frac{1}{M} \sum_{i=1}^M \delta(\theta^{(i)}_t)$ in the following for simplicity. We investigate two types of particle-approximation methods in the following, discrete gradient flows and by blob methods.

**Particle approximation by discrete gradient flows** Denote $\mathcal{P}_2(\mathbb{R}^d)$ be the space of probability measures with finite 2nd-order moments. Define the following optimization problem with stepsize $h$:

$$J_h(\mu) \triangleq \arg \min_{\nu \in \mathcal{P}_2(\mathbb{R}^d)} \left\{ \frac{1}{2h} \mathcal{W}_2^2(\mu, \nu) + E(\nu) \right\}. $$  \hfill (15)

A discrete gradient flow of the continuous one in (15) up to time $T$ is the composition of a sequence of the solutions $(\tilde{\mu}_\ell)_{\ell=1}^T$ of (15), i.e.,

$$\tilde{\mu}_\ell \triangleq J_h(J_h(\cdots J_h(\mu_0))) \triangleq J_{h^\ell}^h \mu_0. $$  \hfill (16)

One can show that when $h \to 0$, the discrete gradient flow (16) converges to the true flow (7) for all $\ell$. Specifically, let $\partial E(\mu)$ be the set of Wasserstein subdifferentiable of $E$ at $\mu$, i.e., $\xi \in \partial E(\mu)$ if $\partial \mu = \xi$ is satisfied. Define $|\partial E(\mu)| = \min\{\|\xi\|_{L^2(\mu)} : \xi \in \partial E(\mu)\}$ to be the minimum norm of the elements in $\partial E(\mu)$. We have

**Lemma 3 (Craig (2014))** Assume $E$ is proper, coercive and lower semicontinuous (specify in Section B of the Supplementary Material (SM)). For an $\mu_0$ and $t \geq 0$, as $\frac{t}{h} \to \infty$, the discrete gradient sequence $\tilde{\mu}_{T/h} \triangleq J_{T/h}^h \mu_0$ converge uniformly in $t$ to a compact subset of $[0, +\infty)$, and $W^2_2(\tilde{\mu}_{T/h}, \mu_T) \leq \sqrt{3} |\partial E(\mu)| \sqrt{T/h}$.

Lemma 3 suggests the discrete gradient flow can approximate the original WGF arbitrarily well if a small enough stepsize $h$ is adopted. Consequently, one solves (16) through a sequence of optimization procedures to update the particles. We will derive a particle-approximation method for the $W_2$ term in (15), which allows us to solve SG-MCMC efficiently. However, this technique is not applicable to SVGD, as we neither have an explicit form of the energy functional in (7) when adopting the $W_2$ metric, nor have an explicit form for the metric $W_{\mathcal{H}}$ in (7) when adopting the KL-divergence as the energy functional. Fortunately, this can be solved by the second approximation method called blob methods.

**Particle approximation by blob methods** The name of blob methods comes from the classical fluids literature, where instead of evolving the density in (7), one evolves all particles on a grid with time-spacing $h$ (Carrillo et al., 2017). Specifically, note the function $v_t$ in (7) represents velocity of particles via transportation map $T$, thus solving a WGF is equivalent to evolving the particles along their velocity in each iteration. Formally, one can prove

**Proposition 4 (Craig and Bertozzi (2016))** Let $\mu_0 \approx \frac{1}{M} \sum_{i=1}^M \delta(\theta_0^{(i)})$. Assume $v_t$ in (7) is well-defined and continuous w.r.t. each $\theta_0^{(i)}$ at time $t$. Then solving the PDE (7) reduces to solving a system of ordinary differential equations for the locations of the Dirac masses:

$$\frac{d\theta_0^{(i)}(t)/dt = -v_t(\theta_0^{(i)}). $$  \hfill (17)

Proposition 4 suggests evolving each particle along the directions defined by $v_t$, eliminating the requirement to know an explicit form of the energy functional. In the following, we apply the above particle-optimization techniques to derive algorithms for SVGD and SG-MCMC.
4.1 A particle-optimization algorithm for SVGD

As mentioned above, discrete-gradient-flow approximation does not apply to SVGD. We thus rely on the blob method. From Section 3, \(v_t(\theta) = (W * \mu_t)(\theta)\). When \(\mu_t(\theta)\) is approximated by particles, \(v_t(\theta)\) is simplified as:

\[
v_t(\theta) = \frac{1}{M} \sum_{j=1}^{M} W(\theta, \theta_j).
\]

As a result, with the definition of \(W\) in \(\Box\), updating \(\{\theta_i\}\) by time discretizing \(\Box\) recovers the update equations for standard SVGD in \(\Box\).

4.2 Particle-optimization algorithms for SG-MCMC

Both the discrete-gradient-flow and the blob methods can be applied for SG-MCMC, which are detailed below.

Particle optimization with discrete gradient flows

We first specify Lemma 5 in the case of SG-MCMC in Lemma 5, which is known as the Jordan-Kinderlehrer-Otto scheme [Jordan et al., 1998].

**Lemma 5 (Jordan et al. [1998])** Assume that \(p(\theta | X)\) is infinitely differentiable, and \(\| \nabla_{\theta} \log p(\theta | X) \| \leq C_3 (1 + C_4 - \log p(\theta | X)) \| \theta \|\) for some constants \(C_3, C_4\). Let \(T = tK\) with \(K\) the number of iterations. \(\mu_t\) be an arbitrary distribution with same support as \(p(\theta | X)\), and \(\{\mu_k\}_{k=1}^{K}\) be the solution of the functional optimization problem:

\[
\mu_k = \arg\min_{\mu \in P_d (\mathbb{R}^d)} KL (\mu || \mu_T) + \frac{1}{2h} W_2^2 (\mu_k - \mu) .
\]

Then \(\mu_k\) converges to \(\mu_T\) in the limit of \(h \to 0\), i.e., \(\lim_{h \to 0} \mu_k = \mu_T\), where \(\mu_T\) is the solution of the FP equation \(\Box\) at time \(T\).

According to Lemma 5, it is apparent that SG-MCMC can be implemented by iteratively solving the optimization problem in \(\Box\). However, particle approximations for both terms in \(\Box\) are challenging. In the following, we develop efficient techniques to solve the problem.

First, rewrite the optimization problem in \(\Box\) as:

\[
\min_{\mu \in P_d (\mathbb{R}^d)} \frac{\mathbb{E} \left[ \log p(\theta | X) \right]}{F_1} - \frac{\mathbb{E} \left[ \log \mu \right]}{F_2} + \frac{1}{2h} W_2^2 (\mu_k - \mu) .
\]

We aim at deriving gradient formulas for both the \(F_1\) and \(F_2\) terms under a particle approximation in order to perform gradient descent for the particles. Let \(\mu \approx \frac{1}{M} \sum_{i=1}^{M} \delta(\theta^{(i)})\). The gradient of \(F_1\) is easily approximated as:

\[
\nabla_{\theta^{(i)}} F_1 = -\nabla_{\theta^{(i)}} \log p(\theta^{(i)} | X) .
\]

To approximate the gradient for \(F_2\), let \(p_{ij}\) denote the joint distribution of the particle-pair \((\theta^{(i)}, \theta^{(j)})\). Note \(\mathbb{E}_\mu[\log \mu]\) is minimized when the particles \(\{\theta^{(i)}\}\) are uniformly distributed. In other words, the marginal distribution vector \((\sum_j p_{ij})\) is a uniform distribution. Combining \(\mathbb{E}_\mu[\log \mu]\) with the definition of \(W_2\), calculating \(F_2\) is equivalent to solving the following optimization problem:

\[
P \triangleq \{p_{ij}\} = \arg\min_{p_{ij}} \sum_{i,j} p_{ij} d_{ij}
\]

subject to \(\sum_j p_{ij} = \frac{1}{M}, \sum_i p_{ij} = \frac{1}{M}\), where \(d_{ij} \triangleq \| \theta^{(i)} - \theta^{(j)} \|^2\). We can further enforce the joint distribution \(\{p_{ij}\}\) to have maximum entropy by introducing a regularization term \(\mathbb{E}_\mu[\log p_{ij}]\), which is stronger than the regularizer enforced for the marginal distribution above. After introducing Lagrangian multipliers \(\{\alpha_i, \beta_i\}\) to deal with the constraints in \(\Box\), we arrive at the dual problem:

\[
\max L^D (\{p_{ij}\}, \{\alpha_i\}, \{\beta_i\}) = \lambda \sum_{i,j} p_{ij} \log p_{ij} + p_{ij} d_{ij}
\]

\[
+ \sum_i \alpha_i \left( \sum_j p_{ij} - \frac{1}{M} \right) + \sum_j \beta_j \left( \sum_i p_{ij} - \frac{1}{M} \right) ,
\]

where \(\lambda\) is the weight for the regularizer. The optimal \(p_{ij}\)'s can be obtained by applying KKT conditions to set the derivative w.r.t. \(p_{ij}\) to be zero, ending up with the following form:

\[
p_{ij}^* = u_i e^{-d_{ij}/\lambda} v_j ,
\]

where \(u_i \triangleq \frac{e^{-\frac{1}{\lambda} \sum_j d_{ij}}}{\sum_j e^{-\frac{1}{\lambda} \sum_j d_{ij}}}\). As a result, the particle gradients on \(F_2\) can be approximated as:

\[
\frac{\partial F_2}{\partial \theta^{(i)}} \approx -\sum_{j} u_i v_j d_{ij} e^{-d_{ij}/\lambda} + 2u_i v_j d_{ij} e^{-d_{ij}/\lambda}
\]

where \(d_{ij} \approx \frac{d_{ij}}{\lambda} - 1\).

Theoretically, we need to adaptively update \(\{u_i, v_j\}\) as well to ensure the constraints in \(\Box\). In practice, however, we use a fixed scaling factor \(\gamma\) to approximate \(u_i v_j\) for the sake of simplicity.

Particle gradients are obtained by combining \(\Box\) and \(\Box\), which are then used to update the particles \(\{\theta^{(i)}\}\) by standard gradient descent. Intuitively, \(\Box\) encourages particles to move to local modes while \(\Box\) regularizes particle interactions. Different from SVGD, our scheme imposes both attractive and repulsive forces for the particles. Specifically, by inspecting \(\Box\), we can conclude that: i) When \(\theta^{(i)}\) is far from a previous particle \(\theta^{(j)}\), i.e., \(d_{ij} > 1\), \(\theta^{(i)}\) is pulled close to \(\theta^{(j)}\) with force proportional to \((d_{ij} - 1)e^{-d_{ij}/\lambda}\); ii) when \(\theta^{(i)}\) is close enough to a previous particle \(\theta^{(j)}\), i.e., \(d_{ij} < 1\), \(\theta^{(i)}\) is pushed away, preventing it from collapsing to \(\theta^{(j)}\).
Particle optimization with blob methods  The idea of blob methods can also be applied to particle approximation for SG-MCMC, which require the velocity vector field \( v_t \). According to [13], this is calculated as:
\[
v_t(\theta) = -\nabla_\theta \frac{\delta(E_1 + E_2)}{\delta \mu} = -\nabla_\theta U - \nabla_\theta \phi / \mu.\]
Unfortunately, direct application of particle approximation is infeasible because the term \( \nabla_\theta \phi \) is undefined with discrete \( \mu \). To tackle this problem, we adopt the idea in [22] to approximate the energy functional \( E_2 \) in [13] as:
\[
E_2 \approx \int \mu(\theta) \log(\mu * K(\theta)) d\theta,\]
where \( K(\cdot, \cdot) \) is another kernel function to smooth out \( \mu \). Consequently, based on [22], the velocity \( v_t \) can be calculated as (details in Section 4 of the SM):
\[
v_t(\theta) = -\nabla_\theta U - \sum_{j=1}^n \nabla_\theta^{(j)} K(\theta, \theta_j^{(j)}) \sum_k K(\theta_j^{(j)}, \theta_t^{(k)}) - \sum_{j=1}^n \nabla_\theta^{(j)} K(\theta, \theta_j^{(j)}) / \sum_{k=1}^n K(\theta, \theta_t^{(k)})\]

Given \( v_t \), particle updates can be obtained by solving (17) numerically as in SVGD. By inspecting the formula of \( v_t \) in (22), the last two terms both act as repulsive forces. Interestingly, the mechanism is similar to SVGD, but with adaptive force between different particle pairs.

5 THE GENERAL RECIPE

Based on the above development, a more general particle-optimization framework is proposed by combining the PDEs of both SG-MCMC and SVGD. As a result, we propose the following PDE to drive evolution of densities
\[
\frac{\partial \mu_t}{\partial t} = -\nabla_\theta \cdot (\mu_t F(\theta_t)) + \lambda_1 \nabla_\theta \cdot ((W * \mu_t) \mu_t) + \lambda_2 \nabla_\theta \cdot g(\theta_t) g^T(\theta_t),\]
where \( \lambda_1 \) and \( \lambda_2 \) are two constants. It is easily seen that to ensure the stationary distribution of (23) to be equal to \( p(\theta \mid X) \), the following condition must be satisfied:
\[
\nabla_\theta \cdot (p(\theta \mid X) F(\theta)) = \lambda_1 \nabla_\theta \cdot ((W * p(\theta \mid X)) p(\theta \mid X)) + \lambda_2 \nabla_\theta \cdot g(\theta_t) g^T(\theta_t)\]

There are many feasible choices for the functions and parameters \( \{F(\theta), W, g(\theta), \lambda_1, \lambda_2\} \) to satisfy (24). However, the verification procedure might be complicated given the present of a convolutional term in (24). We recommend the following choices for simplicity:

- \( F(\theta) = 0, g(\theta) = 0, W \) is defined as (2), this reduces to the Wasserstein-based SGLD with particle optimization. Specifically, when the discrete-gradient-flow approximation is adopted, the algorithm is denoted as \( w \)-SGLD; whereas when the blob method is adopted, it is denoted as \( w \)-SGLD-B.
- \( F(\theta) = \frac{1}{2} U(\theta), W = 0, g(\theta) = 1 \) and \( \lambda_2 = 1 \): this reduces to the Wasserstein-based SGLD with particle optimization.
- \( F(\theta) = \frac{1}{2} U(\theta), g(\theta) = 1 \), \( W \) is defined as (2), and \( \lambda_2 = 1 \): this is the combination of SGLD and SVGD, and is called particle interactive SGLD, denoted as PI-SGLD or \( \pi \)-SGLD.

It is easy to verify that condition (24) is satisfied for all the above three particle-optimization algorithms. Furthermore, particle updates are readily developed by applying either the discrete-gradient-flow or blob-based methods.

6 RELATED PARTICLE-BASED MCMC METHODS

There have been related particle-based MCMC algorithms. Representative methods are sequential Monte Carlo (SMC) (Moral et al., 2006), particle MCMC (PM-MCMC) (Andrieu et al., 2010) and many variants. In SMC, particles are sample from a proposal distribution, and the corresponding weights are updated by a resampling step. PMCMC extends SMC by sampling from an extended distribution interacted with a MH-rejection step. Compared to our framework, their proposal distributions are typically hard to choose; furthermore, optimality of the particles from both methods cannot be guaranteed. Furthermore, the methods are typically much more computationally expensive. Recently, Dai et al. (2016) proposed a particle-based MCMC algorithm by approximating a target distribution with weighted kernel density estimator, which updates particle weights based on likelihoods of the corresponding particles. This approach is theoretically sound but lacks an underlying geometry interpretation. Finally, we note that \( w \)-SGLD has been successfully applied to reinforcement learning recently for improved policy optimization (Zhang et al., 2018a).

7 EXPERIMENTS

We verify our framework on a set of experiments, including a number of toy experiments and applications to Bayesian sampling of deep neural networks (DNNs).

7.1 Demonstrations

Toy Distributions  We compare various sampling methods on multi-mode toy examples, i.e., SGLD, SVGD, \( w \)-SGLD, \( w \)-SGLD-B and \( \pi \)-SGLD. We aim to sample from four unnormalized 2D densities \( p(z) \propto \exp(U(z)) \), with detailed functional form provided in the SM. We optimize/sample 2000 particles to approximate target distributions. The results are shown in Figure 4. It can be seen from Figure 4 that though SGLD maintains good asymptotic properties, it is inaccurate to approximate distributions with only a few samples; in some case, the
samples cannot even cover all the modes. Interestingly, all other particle-optimization-based algorithms successfully find all the modes and fit the distributions well. \(w\)-SGLD is good at finding modes, but worse at modeling the correct variance due to difficulty of controlling the balance between attractive and repulsive forces between particles. \(w\)-SGLD-B is better than \(w\)-SGLD at modeling the distribution variance, performing similarly to SVGD and \(\pi\)-SGLD. Even though, we note that \(w\)-SGLD is very useful when the number of particles is small, which fits a distribution better, as shown in Section E of the SM.

![Figure 1: Illustration of different algorithms on toy distributions. Each column is a distribution case. 1st row: Ground truth; 2nd row: standard SGLD; 3rd row: \(w\)-SGLD; 4th row: \(w\)-SGLD-B; 5th row: SVGD; 6th row: \(\pi\)-SGLD.](image)

**Bayesian Logistic regression** We next compare the three variants of our framework (i.e., SVGD, \(w\)-SGLD and \(w\)-SGLD-B) on a simple logistic-regression task with quantitative evaluations. We use the same model, data and experimental settings as Liu and Wang (2016a). The Covertype dataset contains 581,012 data points and 54 features. We perform 5 runs for each setting and report the mean of testing accuracies/log-likelihoods. Figure 2 plots both test accuracies and test log-likelihoods w.r.t. the number of training iterations. It is clearly that while all methods converge to the same accuracy/likelihood level, both \(w\)-SGLD and \(w\)-SGLD-B converge slightly faster than SVGD. In addition, \(w\)-SGLD and \(w\)-SGLD-B have similar convergence behaviors, thus we only use \(w\)-SGLD in the DNN experiments below.

![Figure 2: Test accuracies (left) and log-likelihoods (right) v.s. iterations for SVGD, \(w\)-SGLD and \(w\)-SGLD-B.](image)

**Parameter Sensitivity**

Now we study the role of hyper-parameters in \(\pi\)-SGLD: the number of particles \(M\) and the scaling factor \(\gamma\) to replace the \(u_i v_j\)-term in (21). We use the same dataset and model as the above experiment. Figure 3 plots test accuracies along with different parameter settings. As expected, the best performance is achieved with appropriate scale of \(W_2^2\). The performance keep improving with increasing particles. Interestingly, the Wasserstein regularization is more important when the number of particles is small, demonstrating the superiority when approximate distributions with very few particles.

![Figure 3: Impact of \(W_2^2\) factor \(\gamma\) and particle number \(M\).](image)

### 7.2 Applications on deep neural networks

We conduct experiments for Bayesian learning of DNNs. Different from traditional optimization for DNNs, we are interested in modeling weight uncertainty of neural networks, an important topic that has been well explored (Hernández-Lobato and Adams, 2015; Blundell et al., 2015a; Li et al., 2016; Louizos and Welling, 2016). We assign priors to the weights, which are simple isotropic Gaussian priors in our case, and perform posterior sampling with the proposed particle-optimization-based algorithms, as well as other standard algorithms such as SGLD and SGD. We use the RMSprop optimizer for feed-forward networks (FNN), and Adam for convolutional neural networks (CNNs) and recurrent neural
networks (RNNs). For all methods, we use a RBF kernel \( K(\theta, \theta') = \exp(-\|\theta - \theta'\|^2/h) \), with the bandwidth set to \( h = \text{med}^2 / \log M \). Here med is the median of the pairwise distance between particles. All experiments are conducted on a single TITAN X GPU.

**Feed-forward Neural Networks** We perform the classification tasks on the standard MNIST dataset. A two-layer model 784-X-10 with ReLU activation function is used, with X being the number of hidden units for each layer. The training epoch is set to 100. The test errors are reported in Table 1. Not surprisingly, Bayesian methods generally perform better than their optimization counterparts. The new \( \pi \)-SGLD which combines \( w \)-SGLD and SVGD improves both methods with little computational overhead. In additional, \( w \)-SGLD seems to perform better than SVGD in this case, partially due to a better asymptotic property mentioned in [Liu, 2017]. Furthermore, standard SGLD which is based on MCMC obtains higher test errors compared to particle-optimization-based algorithms, partially due to the correlated-sample issue discussed in the introduction. See [Blundell et al., 2015b] for details on the other methods in Table 1.

**Convolutional Neural Networks** We use the CIFAR-10 dataset to test our framework on CNNs. We adopt a CNN of three convolution layers, using \( 3 \times 3 \) filter size with C64-C128-C256 channels, and \( 2 \times 2 \) max-pooling after each convolution layer. Our implementation adopts batch normalization, drop out and data augmentation to improve the performance. Training losses and test accuracies are presented in Table 2. Consistently, \( \pi \)-SGLD outperforms all other algorithms in terms of test accuracy. ADAM obtains a better training loss but worse test accuracy, indicating worse generalization ability of the optimization-based methods compared to Bayesian methods.

**Recurrent Neural Networks** For RNNs, we run standard language models. Experiments are presented on three publicly available corpora: APNEWS, IMDB and BNC. APNEWS is a collection of Associated Press news articles from 2009 to 2016. IMDB is a set of movie reviews collected by [Maas et al., 2011], and BNC [BNC Consortium, 2007] is the written portion of the British National Corpus, which contains excerpts from journals, books, letters, essays, memoranda, news and other types of text. These datasets can be downloaded from Github.

![Image of Table 1: Classification error of FNN on MNIST.](https://github.com/jhlau/topically-driven-language-model)

<table>
<thead>
<tr>
<th>Method</th>
<th>Training Loss</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \pi )-SGLD</td>
<td>1.36%</td>
<td>1.30%</td>
</tr>
<tr>
<td>( w )-SGLD</td>
<td>1.44%</td>
<td>1.37%</td>
</tr>
<tr>
<td>SVGD</td>
<td>1.53%</td>
<td>1.40%</td>
</tr>
<tr>
<td>SGLD</td>
<td>1.64%</td>
<td>1.41%</td>
</tr>
<tr>
<td>RMSprop</td>
<td>1.59%</td>
<td>1.43%</td>
</tr>
<tr>
<td>RMSspectral</td>
<td>1.65%</td>
<td>1.56%</td>
</tr>
<tr>
<td>SGD</td>
<td>1.72%</td>
<td>1.47%</td>
</tr>
<tr>
<td>BPB, Gaussian</td>
<td>1.82%</td>
<td>1.99%</td>
</tr>
<tr>
<td>SGD, dropout</td>
<td>1.51%</td>
<td>1.33%</td>
</tr>
</tbody>
</table>

![Image of Table 2: Classification error of CNN on CIFAR-10.](https://github.com/jhlau/topically-driven-language-model)

<table>
<thead>
<tr>
<th>Method</th>
<th>APNEWS</th>
<th>IMDB</th>
<th>BNC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADAM</td>
<td>23.80</td>
<td>86.76%</td>
<td></td>
</tr>
<tr>
<td>SVGD</td>
<td>30.57</td>
<td>88.72%</td>
<td></td>
</tr>
<tr>
<td>SGLD</td>
<td>28.52</td>
<td>88.64%</td>
<td></td>
</tr>
<tr>
<td>( w )-SGLD</td>
<td>31.26</td>
<td>88.80%</td>
<td></td>
</tr>
<tr>
<td>( \pi )-SGLD</td>
<td>25.06</td>
<td>89.52%</td>
<td></td>
</tr>
</tbody>
</table>

![Image of Table 3: Perplexity of language model on three corpora.](https://github.com/jhlau/topically-driven-language-model)

<table>
<thead>
<tr>
<th>Method</th>
<th>APNEWS</th>
<th>IMDB</th>
<th>BNC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGD</td>
<td>64.13</td>
<td>72.14</td>
<td>102.89</td>
</tr>
<tr>
<td>SGLD</td>
<td>63.01</td>
<td>68.12</td>
<td>95.13</td>
</tr>
<tr>
<td>SVGD</td>
<td>61.64</td>
<td>69.25</td>
<td>94.99</td>
</tr>
<tr>
<td>( w )-SGLD</td>
<td>61.22</td>
<td>67.41</td>
<td>93.68</td>
</tr>
<tr>
<td>( \pi )-SGLD</td>
<td>59.83</td>
<td>67.04</td>
<td>92.33</td>
</tr>
</tbody>
</table>

We follow the standard set up as [Wang et al., 2017]. Specifically, we lower case all the word tokens and filter out word tokens that occur less than 10 times. All the datasets are divided into training, development and testing sets. For the language model set up, we consider a 1-layer LSTM model with 600 hidden units. The sequence length is fixed to be 30. In order to alleviate overfitting, dropout with a rate of 0.4 is used in each LSTM layer. Results in terms of test perplexities are presented in Table 3. Again, we see that \( \pi \)-SGLD performs best among all algorithms, and \( w \)-SGLD is slightly better than SVGD, both of which are better than other algorithms.

8 CONCLUSION

We propose a unified particle-optimization framework for large-scale Bayesian sampling. Our framework defines gradient flows on the space of probability measures, and uses particles to approximate the corresponding densities. Consequently, solving gradient flows reduces to optimizing particles on the parameter space. Our framework includes the standard SVGD as a special case, and also allows us to develop efficient particle-optimization algorithms for SG-MCMC, which is highly related to SVGD. Extensive experiments are conducted, demonstrating the effectiveness and efficiency of our proposed framework. Interesting future work includes designing more practically efficient variants of the proposed particle-optimization framework, and developing theory to study general convergence behaviors of the algorithms, in addition to the asymptotic results presented in [Liu, 2017].

[https://github.com/jhlau/topically-driven-language-model]
REFERENCES


