Stein Variational Adaptive Importance Sampling

Jun Han **Qiang Liu**
Computer Science, Dartmouth College, Hanover, NH 03755
{jun.han.gr, qiang.liu}@dartmouth.edu

Abstract

We propose a novel adaptive importance sampling algorithm which incorporates Stein variational gradient decent algorithm (SVGD) with importance sampling (IS). Our algorithm leverages the nonparametric transforms in SVGD to iteratively decrease the KL divergence between importance proposals and target distributions. The advantages of our algorithm are twofold: 1) it turns SVGD into a standard IS algorithm, allowing us to use standard diagnostic and analytic tools of IS to evaluate and interpret the results, and 2) it does not restrict the choice of the importance proposals to predefined distribution families like traditional (adaptive) IS methods. Empirical experiments demonstrate that our algorithm performs well on evaluating partition functions of restricted Boltzmann machines and testing likelihood of variational auto-encoders.

1 INTRODUCTION

Probabilistic modeling provides a fundamental framework for reasoning under uncertainty and modeling complex relations in machine learning. A critical challenge, however, is to develop efficient computational techniques for approximating complex distributions. Specifically, given a complex distribution $p(\boldsymbol{x})$, often known only up to a normalization constant, we are interested estimating integral quantities $\mathbb{E}_p[f]$ for test functions f. Popular approximation algorithms include particle-based methods, such as Monte Carlo, which construct a set of independent particles $\{\boldsymbol{x}_i\}_{i=1}^n$ whose empirical averaging $\frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)$ forms unbiased estimates of $\mathbb{E}_p[f]$, and variational inference (VI), which approximates p with a simpler surrogate distribution q by minimizing a KL divergence objective function within a predefined parametric family of distributions. Modern variational inference methods have found successful applications in highly complex learning systems (e.g., Hoffman et al., 2013; Kingma & Welling, 2013). However, VI critically depends on the choice of parametric families and does not generally provide consistent estimators like particle-based methods.

Stein variational gradient descent (SVGD) is an alternative framework that integrates both the particle-based and variational ideas. It starts with a set of initial particles $\{\boldsymbol{x}_i^0\}_{i=1}^n$, and iteratively updates the particles using adaptively constructed deterministic variable transforms:

$$\boldsymbol{x}_i^\ell \leftarrow \boldsymbol{T}_\ell(\boldsymbol{x}_i^{\ell-1}), \quad \forall i = 1, \ldots, n,$$

where $\boldsymbol{T}_\ell$ is a variable transformation at the ℓ-th iteration that maps old particles to new ones, constructed adaptively at each iteration based on the most recent particles $\{\boldsymbol{x}_i^{\ell-1}\}_{i=1}^n$ that guarantee to push the particles "closer" to the target distribution p, in the sense that the KL divergence between the distribution of the particles and the target distribution p can be iteratively decreased. More details on the construction of $\boldsymbol{T}_\ell$ can be found in Section 2.

In the view of measure transport, SVGD iteratively transports the initial probability mass of the particles to the target distribution. SVGD constructs a path of distributions that bridges the initial distribution q_0 to the target distribution p,

$$q_\ell = (\boldsymbol{T}_\ell \circ \cdots \circ \boldsymbol{T}_1)\sharp q_0, \quad \ell = 1, \ldots, K. \tag{1}$$

where $\boldsymbol{T}\sharp q$ denotes the push-forward measure of q through the transform $\boldsymbol{T}$, that is the distribution of $\boldsymbol{z} = \boldsymbol{T}(\boldsymbol{x})$ when $\boldsymbol{x} \sim q$.

The story, however, is complicated by the fact that the transform $\boldsymbol{T}_\ell$ is practically constructed on the fly *depending* on the recent particles $\{\boldsymbol{x}_i^{\ell-1}\}_{i=1}^n$, which introduces complex dependency between the particles at the next iteration, whose theoretical understanding requires mathematical tools in interacting particle systems (e.g., Braun & Hepp, 1977; Spohn, 2012; Del Moral, 2013) and propagation of chaos (e.g., Sznitman, 1991). As a result, $\{\boldsymbol{x}_i^\ell\}_{i=1}^n$ can not be viewed as i.i.d. samples from q_ℓ. This makes it difficult to analyze the results of SVGD and quantify their bias and variance.

In this paper, we propose a simple modification of SVGD that "decouples" the particle interaction and returns particles i.i.d. drawn from q_ℓ; we also develop a method to iteratively keep track of the importance weights of these particles, which makes it possible to give consistent, or unbiased estimators within finite number of iterations of SVGD.

Our method integrates SVGD with importance sampling (IS) and combines their advantages: it leverages the SVGD dynamics to obtain high quality proposals q_ℓ for IS and also turns SVGD into a standard IS algorithm, inheriting the interpretability and theoretical properties of IS. Another advantage of our proposed method is that it provides an SVGD-based approach for estimating intractable normalization constants, an inference problem that the original SVGD does not offer to solve.

Related Work Our method effectively turns SVGD into a nonparametric, adaptive importance sampling (IS) algorithm, where the importance proposal q_ℓ is adaptively improved by the optimal transforms $\boldsymbol{T}_\ell$ which maximally decreases the KL divergence between the iterative distribution and the target distribution in a function space. This is in contrast to the traditional adaptive importance sampling methods (e.g., Cappé et al., 2008; Ryu & Boyd, 2014; Cotter et al., 2015), which optimize the proposal distribution from predefined distribution families $\{q_{\boldsymbol{\theta}}(\boldsymbol{x})\}$, often mixture families or exponential families. The parametric assumptions restrict the choice of the proposal distributions and may give poor results when the assumption is inconsistent with the target distribution p. The proposals q_ℓ in our method, however, are obtained by recursive variable transforms constructed in a nonparametric fashion and become more complex as more transforms $\boldsymbol{T}_\ell$ are applied. In fact, one can view q_ℓ as the result of pushing q_0 through a neural network with ℓ-layers, constructed in a non-parametric, layer-by-layer fashion, which provides a much more flexible distribution family than typical parametric families such as mixtures or exponential families.

There has been a collection of recent works (such as Rezende & Mohamed, 2015; Kingma et al., 2016; Marzouk et al., 2016; Spantini et al., 2017), that approximate the target distributions with complex proposals obtained by iterative variable transforms in a similar way to our proposals q_ℓ in (1). The key difference, however, is that these methods explicitly parameterize the transforms $\boldsymbol{T}_\ell$ and optimize the parameters by back-propagation, while our method, by leveraging the nonparametric nature of SVGD, constructs the transforms $\boldsymbol{T}_\ell$ sequentially in closed forms, requiring no back-propagation.

The idea of constructing a path of distributions $\{q_\ell\}$ to bridge the target distribution p with a simpler distribution q_0 invites connection to ideas such as annealed importance sampling (AIS) (Neal, 2001) and path sampling (PS) (Gelman & Meng, 1998). These methods typically construct an annealing path using geometric averaging of the initial and target densities instead of variable transforms, which does not build in a notion of variational optimization as the SVGD path. In addition, it is often intractable to directly sample distributions on the geometry averaging path, and hence AIS and PS need additional mechanisms in order to construct proper estimators.

Outlines The reminder of this paper is organized as follows. Section 2 discusses Stein discrepancy and SVGD. We propose our main algorithm in Section 3, and a related method in Section 4. Section 5 provides empirical experiments and Section 6 concludes the paper.

2 STEIN VARIATIONAL GRADIENT DESCENT

We introduce the basic idea of Stein variational gradient descent (SVGD) and Stein discrepancy. The readers are referred to Liu & Wang (2016) and Liu et al. (2016) for more detailed introduction.

Preliminary We always assume $\boldsymbol{x} = [x_1, \cdots, x_d]^\top \in \mathbb{R}^d$ in this paper. Given a positive definite kernel $k(\boldsymbol{x}, \boldsymbol{x}')$, there exists an unique reproducing kernel Hilbert space (RKHS) $\mathcal{H}_0$, formed by the closure of functions of form $f(\boldsymbol{x}) = \sum_i a_i k(\boldsymbol{x}, \boldsymbol{x}_i)$ where $a_i \in \mathbb{R}$, equipped with inner product $\langle f,\ g\rangle_{\mathcal{H}_0} = \sum_{ij} a_i k(\boldsymbol{x}_i, \boldsymbol{x}_j) b_j$ for $g(\boldsymbol{x}) = \sum_j b_j k(\boldsymbol{x}, \boldsymbol{x}_j)$. Denote by $\mathcal{H} = \mathcal{H}_0^d = \mathcal{H}_0 \times \cdots \times \mathcal{H}_0$ the vector-valued function space formed by $\boldsymbol{f} = [f_1, \ldots, f_d]^\top$, where $f_i \in \mathcal{H}_0$, $i = 1, \ldots, d$, equipped with inner product $\langle \boldsymbol{f},\ \boldsymbol{g}\rangle_{\mathcal{H}} = \sum_{l=1}^d \langle f_l,\ g_l\rangle_{\mathcal{H}_0}$, for $\boldsymbol{g} = [g_1, \ldots, g_d]^\top$. Equivalently, $\mathcal{H}$ is the closure of functions of form $\boldsymbol{f}(\boldsymbol{x}) = \sum_i \boldsymbol{a}_i k(\boldsymbol{x}, \boldsymbol{x}_i)$ where $\boldsymbol{a}_i \in \mathbb{R}^d$ with inner product $\langle \boldsymbol{f},\ \boldsymbol{g}\rangle_{\mathcal{H}} = \sum_{ij} \boldsymbol{a}_i^\top \boldsymbol{b}_j k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for $\boldsymbol{g}(\boldsymbol{x}) = \sum_i \boldsymbol{b}_i k(\boldsymbol{x}, \boldsymbol{x}_i)$. See e.g., Berlinet & Thomas-Agnan (2011) for more background on RKHS.

2.1 Stein Discrepancy as Gradient of KL Divergence

Let $p(\boldsymbol{x})$ be a density function on $\mathbb{R}^d$ which we want to approximate. We assume that we know $p(\boldsymbol{x})$ only up to a normalization constant, that is,

$$p(\boldsymbol{x}) = \frac{1}{Z}\bar{p}(\boldsymbol{x}), \qquad Z = \int \bar{p}(\boldsymbol{x}) d\boldsymbol{x}, \tag{2}$$

where we assume we can only calculate $\bar{p}(\boldsymbol{x})$ and Z is a normalization constant (known as the partition function) that is intractable to calculate exactly. We assume that $\log p(\boldsymbol{x})$ is differentiable w.r.t. $\boldsymbol{x}$, and we have access to $\nabla \log p(\boldsymbol{x}) = \nabla \log \bar{p}(\boldsymbol{x})$ which does not depend on Z.

The main idea of SVGD is to use a set of sequential deterministic transforms to iteratively push a set of particles

$\{\boldsymbol{x}_i\}_{i=1}^n$ towards the target distribution:

$$\begin{aligned}&\boldsymbol{x}_i \leftarrow \boldsymbol{T}(\boldsymbol{x}_i), \qquad \forall i = 1, 2, \cdots, n\\ &\boldsymbol{T}(\boldsymbol{x}) = \boldsymbol{x} + \epsilon\boldsymbol{\phi}(\boldsymbol{x}),\end{aligned} \tag{3}$$

where we choose the transform $\boldsymbol{T}$ to be an additive perturbation by a velocity field $\boldsymbol{\phi}$, with a magnitude controlled by a step size ϵ that is assumed to be small.

The key question is the choice of the velocity field $\boldsymbol{\phi}$; this is done by choosing $\boldsymbol{\phi}$ to maximally decrease the KL divergence between the distribution of particles and the target distribution. Assume the current particles are drawn from q, and $\boldsymbol{T}\sharp q$ is the distribution of the updated particles, that is, $\boldsymbol{T}\sharp q$ is the distribution of $\boldsymbol{x}' = \boldsymbol{T}(\boldsymbol{x}) = \boldsymbol{x} + \epsilon\boldsymbol{\phi}(\boldsymbol{x})$ when $\boldsymbol{x} \sim q$. The optimal $\boldsymbol{\phi}$ should solve the following functional optimization:

$$\mathbb{D}(q \,||\, p) \overset{def}{=} \max_{\boldsymbol{\phi} \in \mathcal{F} \colon ||\boldsymbol{\phi}||_{\mathcal{F}} \le 1} \left\{ -\frac{d}{d\epsilon}\mathrm{KL}(\boldsymbol{T}\sharp q \,||\, p) \,\big|_{\epsilon=0} \right\}, \tag{4}$$

where $\mathcal{F}$ is a vector-valued normed function space that contains the set of candidate velocity fields $\boldsymbol{\phi}$.

The maximum negative gradient value $\mathbb{D}(q \,||\, p)$ in (4) provides a discrepancy measure between two distributions q and p and is known as *Stein discrepancy* (Gorham & Mackey, 2015; Liu et al., 2016; Chwialkowski et al., 2016): if $\mathcal{F}$ is taken to be large enough, we have $\mathbb{D}(q \,||\, p) = 0$ iff there exists no transform to further improve the KL divergence between p and q, namely $p = q$.

It is necessary to use an infinite dimensional function space $\mathcal{F}$ to obtain good transforms, which then casts a challenging functional optimization problem. Fortunately, it turns out that a simple closed form solution can be obtained by taking $\mathcal{F}$ to be an RKHS $\mathcal{H} = \mathcal{H}_0 \times \cdots \mathcal{H}_0$, where $\mathcal{H}_0$ is a RKHS of scalar-valued functions, associated with a positive definite kernel $k(x, x')$. In this case, Liu et al. (2016) showed that the optimal solution of (4) is $\boldsymbol{\phi}^*/||\boldsymbol{\phi}^*||_{\mathcal{H}}$, where

$$\boldsymbol{\phi}^*(\cdot) = \mathbb{E}_{\boldsymbol{x}\sim q}[\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}) k(\boldsymbol{x}, \cdot) + \nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \cdot)]. \tag{5}$$

In addition, the corresponding Stein discrepancy, known as kernelized Stein discrepancy (KSD) (Liu et al., 2016; Chwialkowski et al., 2016; Gretton et al., 2009; Oates et al., 2016), can be shown to have the following closed form

$$\mathbb{D}(q \,||\, p) = ||\boldsymbol{\phi}^*||_{\mathcal{H}} = \big(\mathbb{E}_{x, x' \sim q}[\kappa_p(\boldsymbol{x}, \boldsymbol{x}')]\big)^{1/2}, \tag{6}$$

where $\kappa_p(x, x')$ is a positive definite kernel defined by

$$\begin{aligned}\kappa_p(\boldsymbol{x}, \boldsymbol{x}') &= \boldsymbol{s}_p(\boldsymbol{x})^\top k(\boldsymbol{x}, \boldsymbol{x}')\boldsymbol{s}_p(\boldsymbol{x}') + \boldsymbol{s}_p(\boldsymbol{x})^\top \nabla_{\boldsymbol{x}'} k(\boldsymbol{x}, \boldsymbol{x}')\\ &+ \boldsymbol{s}_p(\boldsymbol{x}')^\top \nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{x}') + \nabla_{\boldsymbol{x}} \cdot (\nabla_{\boldsymbol{x}'} k(\boldsymbol{x}, \boldsymbol{x}')).\end{aligned} \tag{7}$$

where $\boldsymbol{s}_p(\boldsymbol{x}) \overset{def}{=} \nabla \log p(\boldsymbol{x})$. We refer to Liu et al. (2016) for the derivation of (7), and further treatment of KSD in Chwialkowski et al. (2016); Oates et al. (2016); Gorham & Mackey (2017).

2.2 Stein Variational Gradient Descent

In order to apply the derived optimal transform in the practical SVGD algorithm, we approximate the expectation $\mathbb{E}_{\boldsymbol{x}\sim q}[\cdot]$ in (5) using the empirical averaging of the current particles, that is, given particles $\{\boldsymbol{x}_i^\ell\}_{i=1}^n$ at the ℓ-th iteration, we construct the following velocity field:

$$\boldsymbol{\phi}_{\ell+1}(\cdot) = \frac{1}{n}\sum_{j=1}^n [\nabla \log p(\boldsymbol{x}_j^\ell) k(\boldsymbol{x}_j^\ell, \cdot) + \nabla_{\boldsymbol{x}_j^\ell} k(\boldsymbol{x}_j^\ell, \cdot)]. \tag{8}$$

The SVGD update at the ℓ-th iteration is then given by

$$\begin{aligned}&\boldsymbol{x}_i^{\ell+1} \leftarrow \boldsymbol{T}_{\ell+1}(\boldsymbol{x}_i^\ell),\\ &\boldsymbol{T}_{\ell+1}(\boldsymbol{x}) = \boldsymbol{x} + \epsilon\boldsymbol{\phi}_{\ell+1}(\boldsymbol{x}).\end{aligned} \tag{9}$$

Here transform $\boldsymbol{T}_{\ell+1}$ is adaptively constructed based on the most recent particles $\{\boldsymbol{x}_i^\ell\}_{i=1}^n$. Assume the initial particles $\{\boldsymbol{x}_i^0\}_{i=1}^n$ are i.i.d. drawn from some distribution q_0, then the pushforward maps of $\boldsymbol{T}_\ell$ define a sequence of distributions that bridges between q_0 and p:

$$q_\ell = (\boldsymbol{T}_\ell \circ \cdots \circ \boldsymbol{T}_1)\sharp q_0, \quad \ell = 1, \ldots, K, \tag{10}$$

where q_ℓ forms increasingly better approximation of the target p as ℓ increases. Because $\{\boldsymbol{T}_\ell\}$ are nonlinear transforms, q_ℓ can represent highly complex distributions even when the original q_0 is simple. In fact, one can view q_ℓ as a deep residual network (He et al., 2016) constructed layer-by-layer in a fast, nonparametric fashion.

However, because the transform $\boldsymbol{T}_\ell$ depends on the previous particles $\{\boldsymbol{x}_i^{\ell-1}\}_{i=1}^n$ as shown in (8), the particles $\{\boldsymbol{x}_i^\ell\}_{i=1}^n$, after the zero-th iteration, depend on each other in a complex fashion, and do not, in fact, straightforwardly follow distribution q_ℓ in (10). Principled approaches for analyzing such interacting particle systems can be found in Braun & Hepp (e.g., 1977); Spohn (e.g., 2012); Del Moral (e.g., 2013); Sznitman (e.g., 1991). The goal of this work, however, is to provide a simple method to "decouple" the SVGD dynamics, transforming it into a standard importance sampling method that is amendable to easier analysis and interpretability, and also applicable to more general inference tasks such as estimating partition function of unnormalized distribution where SVGD cannot be applied.

3 DECOUPLING SVGD

In this section, we introduce our main Stein variational importance sampling (SteinIS) algorithm. Our idea is simple. We initialize the particles $\{\boldsymbol{x}_i^0\}_{i=1}^n$ by i.i.d. draws from an initial distribution q_0 and partition them into two sets, including a set of *leader particles* $\boldsymbol{x}_A^\ell = \{\boldsymbol{x}_i^\ell \colon i \in A\}$ and *follower particles* $\boldsymbol{x}_B^\ell = \{\boldsymbol{x}_i^\ell \colon i \in B\}$, with $B = \{1, \ldots, n\} \setminus A$, where the leader particles $\boldsymbol{x}_A^\ell$ are responsible for constructing the transforms, using the standard

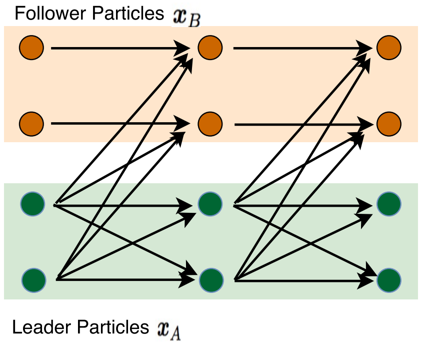

Follower Particles $\boldsymbol{x}_B$
Leader Particles $\boldsymbol{x}_A$

Figure 1: Our method uses a set of leader particles $\boldsymbol{x}_A^\ell$ (green) to construct the transform map $\boldsymbol{T}_\ell$, which follower particles $\boldsymbol{x}_B^\ell$ follows subsequently. The leader particles $\boldsymbol{x}_A^\ell$ are interactive and dependent on each other. The follower particles $\boldsymbol{x}_B^\ell$ can be viewed as i.i.d. draws from q_ℓ, given fixed leader particles $\boldsymbol{x}_A^\ell$.

SVGD update (9), while the follower particles $\boldsymbol{x}_B^\ell$ simply follow the transform maps constructed by $\boldsymbol{x}_A^\ell$ and do not contribute to the construction of the transforms. In this way, the follower particles $\boldsymbol{x}_B^\ell$ are independent conditional on the leader particles $\boldsymbol{x}_A^\ell$.

Conceptually, we can think that we first construct all the maps $\boldsymbol{T}_\ell$ by evolving the leader particles $\boldsymbol{x}_A^\ell$, and then push the follower particles through $\boldsymbol{T}_\ell$ in order to draw exact, i.i.d. samples from q_ℓ in (10). Note that this is under the assumption the leader particles $\boldsymbol{x}_A^\ell$ has been observed and fixed, which is necessary because the transform $\boldsymbol{T}_\ell$ and distribution q_ℓ depend on $\boldsymbol{x}_A^\ell$.

In practice, however, we can simultaneously update both the leader and follower particles, by a simple modification of the original SVGD (9) shown in Algorithm 1 (step 1-2), where the only difference is that we restrict the empirical averaging in (8) to the set of the leader particles $\boldsymbol{x}_A^\ell$. The relationship between the particles in set A and B can be more easily understood in Figure 1.

Calculating the Importance Weights Because q_ℓ is still different from p when we only apply finite number of iterations ℓ, which introduces deterministic biases if we directly use $\boldsymbol{x}_B^\ell$ to approximate p. We address this problem by further turning the algorithm into an importance sampling algorithm with importance proposal q_ℓ. Specifically, we calculate the importance weights of the particles $\{\boldsymbol{x}_i^\ell\}$:

$$w_i^\ell = \frac{\bar{p}(\boldsymbol{x}_i^\ell)}{q_\ell(\boldsymbol{x}_i^\ell)}, \tag{11}$$

where $\bar{p}$ is the unnormalized density of p, that is, $p(\boldsymbol{x}) = \bar{p}(\boldsymbol{x})/Z$ as in (2). In addition, the importance weights in (11) can be calculated based on the following formula:

$$q_\ell(\boldsymbol{x}^\ell) = q_0(\boldsymbol{x}^0) \prod_{\jmath=1}^{\ell} |\det(\nabla_{\boldsymbol{x}} \boldsymbol{T}_\jmath(\boldsymbol{x}^{\jmath-1}))|^{-1}, \tag{12}$$

where $\boldsymbol{T}_\ell$ is defined in (9) and we assume that the step size ϵ is small enough so that each $\boldsymbol{T}_\ell$ is an one-to-one map. As shown in Algorithm 1 (step 3), (12) can be calculated recursively as we update the particles.

Algorithm 1 Stein Variational Importance Sampling

```
**Goal**: Obtain i.i.d. importance sample $\{\boldsymbol{x}_i^K,\ w_i^K\}$ for $p$.
Initialize $\boldsymbol{x}_A^0$ and $\boldsymbol{x}_B^0$ by i.i.d. draws from $q_0$.
Calculate $\{q_0(\boldsymbol{x}_i^0)\}, \forall i \in B$.
**for** iteration $\ell = 0, \ldots, K-1$ **do**
  1. Construct the map using the leader particles $\boldsymbol{x}_A^\ell$
     $\boldsymbol{\phi}_{\ell+1}(\cdot) = \frac{1}{|A|} \sum_{j \in A} [\nabla \log p(\boldsymbol{x}_j^\ell) k(\boldsymbol{x}_j^\ell, \cdot) + \nabla_{\boldsymbol{x}_j^\ell} k(\boldsymbol{x}_j^\ell, \cdot)].$
  2. Update both the leader and follower particles
     $\boldsymbol{x}_i^{\ell+1} \leftarrow \boldsymbol{x}_i^\ell + \epsilon \boldsymbol{\phi}_{\ell+1}(\boldsymbol{x}_i^\ell), \quad \forall i \in A \cup B.$
  3. Update the density values (for $i \in B$) by
     $q_{\ell+1}(\boldsymbol{x}_i^{\ell+1}) = q_\ell(\boldsymbol{x}_i^\ell) \cdot |\det(I \ + \ \epsilon \nabla_{\boldsymbol{x}} \boldsymbol{\phi}_{\ell+1}(\boldsymbol{x}_i^\ell))|^{-1}$
**end for**
Calcuate $w_i^K = p(\boldsymbol{x}_i^K)/q_K(\boldsymbol{x}_i^K), \forall i \in B$.
**Outputs**: i.i.d. importance sample $\{\boldsymbol{x}_i^K,\ w_i^K\}$ for $i \in B$.
```

With the importance weights calculated, we turn SVGD into a standard importance sampling algorithm. For example, we can now estimate expectations of form $\mathbb{E}_p f$ by

$$\hat{\mathbb{E}}_p[f] = \frac{\sum_{i \in B} w_i^\ell f(\boldsymbol{x}_i^\ell)}{\sum_{i \in B} w_i^\ell},$$

which provides a consistent estimator of $\mathbb{E}_p f$ when we use finite number ℓ of transformations. Here we use the self normalized weights because $\bar{p}(\boldsymbol{x})$ is unnormalized. Further, the sum of the unnormalized weights provides an unbiased estimation for the normalization constant Z:

$$\hat{Z} = \frac{1}{|B|} \sum_{i \in B} w_i^\ell,$$

which satisfies the unbiasedness property $\mathbb{E}[\hat{Z}] = Z$. Note that the original SVGD does not provide a method for estimating normalization constants, although, as a side result of this work, Section 4 will discuss another method for estimating Z that is more directly motivated by SVGD.

We now analyze the time complexity of our algorithm. Let $\alpha(d)$ be the cost of computing $\boldsymbol{s}_p(\boldsymbol{x})$ and $\beta(d)$ be the cost of evaluating kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ and its gradient $\nabla k(\boldsymbol{x}, \boldsymbol{x}')$. Typically, both $\alpha(d)$ and $\beta(d)$ grow linearly with the dimension d. In most cases, $\alpha(d)$ is much larger than $\beta(d)$. The complexity of the original SVGD with $|A|$ particles is $O(|A|\alpha(d) + |A|^2\beta(d))$, and the complexity of Algorithm 1 is $O(|A|\alpha(d) + |A|^2\beta(d) + |B||A|\beta(d) + |B|d^3)$, where the $O(|B|d^3)$ complexity comes from calculating the determinant of the Jacobian matrix, which is expensive when dimension d is high, but is the cost to pay for having a consistent importance sampling estimator in finite iterations and for being able to estimate the normalization constant Z. Also, by calculating the effective sample size based on the importance weights, we can assess the accuracy of the estimator, and early stop the algorithm when a confidence threshold is reached.

One way to speed up our algorithm in empirical experiments is to parallelize the computation of Jacobian matrices for all follower particles in GPU. It is possible, however, to develop efficient approximation for the determinants by leveraging the special structure of the Jacobean matrix; note that

$$\nabla_{\boldsymbol{y}}\boldsymbol{T}(\boldsymbol{y}) = I + \epsilon A,$$
$$A = \frac{1}{n}\sum_{j=1}^{n}[\nabla_{\boldsymbol{x}}\log p(\boldsymbol{x}_j)^\top \nabla_{\boldsymbol{y}} k(\boldsymbol{x}_j, \boldsymbol{y}) + \nabla_{\boldsymbol{x}}\nabla_{\boldsymbol{y}} k(\boldsymbol{x}_j, \boldsymbol{y})].$$

Therefore, $\nabla_{\boldsymbol{y}}\boldsymbol{T}(\boldsymbol{y})$ is close to the identity matrix I when the step size is small. This allows us to use Taylor expansion for approximation:

Proposition 1. *Assume $\epsilon < 1/\rho(A)$, where $\rho(A)$ is the spectral radius of A, that is, $\rho(A) = \max_j |\lambda_j(A)|$ and $\{\lambda_j\}$ are the eigenvalues of A. We have*

$$\det(I + \epsilon A) = \prod_{k=1}^{d}(1 + \epsilon a_{kk}) + O(\epsilon^2), \qquad (13)$$

where $\{a_{kk}\}$ are the diagonal elements of A.

Proof. Use the Taylor expansion of $\det(I + \epsilon A)$. □

Therefore, one can approximate the determinant with approximation error $O(\epsilon^2)$ using linear time $O(d)$ w.r.t. the dimension. Often the step size is decreasing with iterations, and a way to trade-off the accuracy with computational cost is to use the exact calculation in the beginning when the step size is large, and switch to the approximation when the step size is small.

3.1 Monotone Decreasing of KL divergence

One nice property of algorithm 1 is that the KL divergence between the iterative distribution q_ℓ and p is monotonically decreasing. This property can be more easily understood by considering our iterative system in continuous evolution time as shown in Liu (2017). Take the step size ϵ of the transformation defined in (3) to be infinitesimal, and define the continuos time $t = \epsilon\ell$. Then the evolution equation of random variable $\boldsymbol{x}^t$ is governed by the following nonlinear partial differential equation (PDE),

$$\frac{d\boldsymbol{x}^t}{dt} = \mathbb{E}_{\boldsymbol{x}\sim q_t}[\boldsymbol{s}_p(\boldsymbol{x})k(\boldsymbol{x}, \boldsymbol{x}^t) + \nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{x}^t)], \qquad (14)$$

where t is the current evolution time and q_t is the density function of $\boldsymbol{x}^t$. The current evolution time $t = \epsilon\ell$ when ϵ is small and ℓ is the current iteration. We have the following proposition (see also Liu (2017)):

Proposition 2. *Suppose random variable $\boldsymbol{x}^t$ is governed by PDE* (14)*, then its density q_t is characterized by*

$$\frac{\partial q_t}{\partial t} = -\mathrm{div}(q_t \mathbb{E}_{\boldsymbol{x}\sim q_t}[\boldsymbol{s}_p(\boldsymbol{x})k(\boldsymbol{x}, \boldsymbol{x}^t) + \nabla_{\boldsymbol{x}} k(\boldsymbol{x}, \boldsymbol{x}^t)]), \qquad (15)$$

where $\mathrm{div}(\boldsymbol{f}) = \mathrm{trace}(\nabla \boldsymbol{f}) = \sum_{i=0}^{d} \partial f_i(\boldsymbol{x})/\partial x_i$, and $\boldsymbol{f} = [f_1, \ldots, f_d]^\top$.

The proof of proposition 2 is similar to the proofs of proposition 1.1 in Jourdain & Méléard (1998). Proposition 2 characterizes the evolution of the density function $q_t(\boldsymbol{x}^t)$ when the random variable $\boldsymbol{x}^t$ is evolved by (14). The continuous system captured by (14) and (15) is a type of Vlasov process which has wide applications in physics, biology and many other areas (e.g., Braun & Hepp, 1977). As a consequence of proposition 2, one can show the following nice property:

$$\frac{d\mathrm{KL}(q_t \,||\, p)}{dt} = -\mathbb{D}(q_t \,||\, p)^2 < 0, \qquad (16)$$

which is proved by theorem 4.4 in Liu (2017). Equation (16) indicates that the KL divergence between the iterative distribution q_t and p is monotonically decreasing with a rate of $\mathbb{D}(q_t \,||\, p)^2$.

4 A PATH INTEGRATION METHOD

We mentioned that the original SVGD does not have the ability to estimate the partition function. Section 3 addressed this problem by turning SVGD into a standard importance sampling algorithm in Section 3. Here we introduce another method for estimating KL divergence and normalization constants that is more directly motivated by the original SVGD, by leveraging the fact that the Stein discrepancy is a type of gradient of KL divergence. This method does not need to estimate the importance weights but has to run SVGD to converge to diminish the Stein discrepancy between intermediate distribution q_ℓ and p. In addition, this method does not perform as well as Algorithm 1 as we find empirically. Nevertheless, we find this idea is conceptually interesting and useful to discuss it.

Recalling Equation (4) in Section 2.1, we know that if we perform transform $\boldsymbol{T}(\boldsymbol{x}) = \boldsymbol{x} + \epsilon\boldsymbol{\phi}^*(\boldsymbol{x})$ with $\boldsymbol{\phi}^*$ defined in (5), the corresponding decrease of KL divergence would

Algorithm 2 SVGD with Path Integration for estimating $\mathrm{KL}(q_0 \,||\, p)$ and $\log Z$

```
Input: Target distribution $p(x) = \bar{p}(x)/Z$; an initial distribution $q_0$.
Goal: Estimating $\mathrm{KL}(q_0 \,||\, p)$ and the normalization constant $\log Z$.
Initialize $\hat{K} = 0$. Initialize particles $\{\boldsymbol{x}_i^0\}_{i=1}^n \sim q_0(\boldsymbol{x})$.
Compute $\hat{\mathbb{E}}_{q_0}[\log(q_0(\boldsymbol{x})/\bar{p}(\boldsymbol{x}))]$ via sampling from $q_0$.
while iteration $\ell$ do
    $\hat{K} \leftarrow \hat{K} + \epsilon \hat{\mathbb{D}}(q_\ell \,||\, p)^2,$
    $\boldsymbol{x}_i^{\ell+1} \leftarrow \boldsymbol{x}_i^\ell + \boldsymbol{\phi}_{\ell+1}(\boldsymbol{x}_i^\ell),$
    where $\hat{\mathbb{D}}(q_\ell \,||\, p)$ is defined in (19).
end while
Estimate $\mathrm{KL}(q_0 \,||\, p)$ by $\hat{K}$ and $\log Z$ by $\hat{\mathbb{D}} - \hat{\mathbb{E}}_{q_0}[\log(q_0(\boldsymbol{x})/\bar{p}(\boldsymbol{x}))]$.
```

be

$$\begin{aligned} \mathrm{KL}(q \,||\, p) - \mathrm{KL}(\boldsymbol{T}\sharp q \,||\, p) &\approx \epsilon \cdot ||\boldsymbol{\phi}^*||_{\mathcal{H}} \cdot \mathbb{D}(q \,||\, p) \\ &\approx \epsilon \cdot \mathbb{D}(q \,||\, p)^2, \end{aligned} \tag{17}$$

where we used the fact that $\mathbb{D}(q \,||\, p) = ||\boldsymbol{\phi}^*||_{\mathcal{H}}$, shown in (6). Applying this recursively on q_ℓ in (17), we get

$$\mathrm{KL}(q_0 \,||\, p) - \mathrm{KL}(q_{\ell+1} \,||\, p) \approx \sum_{\jmath=0}^{\ell} \epsilon \cdot \mathbb{D}(q_\jmath \,||\, p)^2.$$

Assuming $\mathrm{KL}(q_\ell \,||\, p) \to 0$ when $\ell \to \infty$, we get

$$\mathrm{KL}(q_0 \,||\, p) \approx \sum_{\ell=0}^{\infty} \epsilon \cdot \mathbb{D}(q_\ell \,||\, p)^2. \tag{18}$$

By (6), the square of the KSD can be empirically estimated via V-statistics, which is given as

$$\hat{\mathbb{D}}(q_\ell \,||\, p)^2 = \frac{1}{n^2} \sum_{i,j=1}^{n} \kappa(\boldsymbol{x}_i^\ell, \boldsymbol{x}_j^\ell). \tag{19}$$

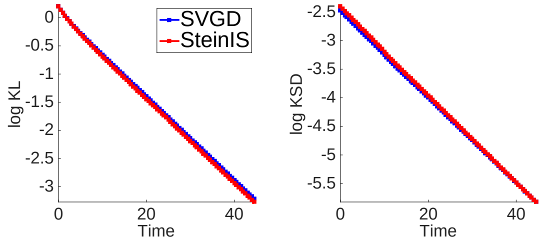
Overall, equation (18) and (19) give an estimator of the KL divergence between q_0 and $p = \bar{p}(\boldsymbol{x})/Z$. This can be transformed into an estimator of the log normalization constant $\log Z$ of p, by noting that

$$\log Z = \mathrm{KL}(q_0 \,||\, p) - \mathbb{E}_{q_0}[\log(q_0(\boldsymbol{x})/\bar{p}(\boldsymbol{x}))], \tag{20}$$

where the second term can be estimated by drawing a lot of samples to diminish its variance since the samples from q_0 is easy to draw. The whole procedure is summarized in Algorithm 2.

5 EMPIRICAL EXPERIMENTS

We study the empirical performance of our proposed algorithms on both simulated and real world datasets. We start with toy examples to numerically investigate some theoretical properties of our algorithms, and compare it with traditional adaptive IS on non-Gaussian, multi-modal distributions. We also employ our algorithm to estimate the partition function of Gaussian-Bernoulli Restricted Boltzmann Machine(RBM), a graphical model widely used in deep learning (Welling et al., 2004; Hinton & Salakhutdinov, 2006), and to evaluate the log likelihood of decoder models in variational autoencoder (Kingma & Welling, 2013).



(a) KL (b) KSD

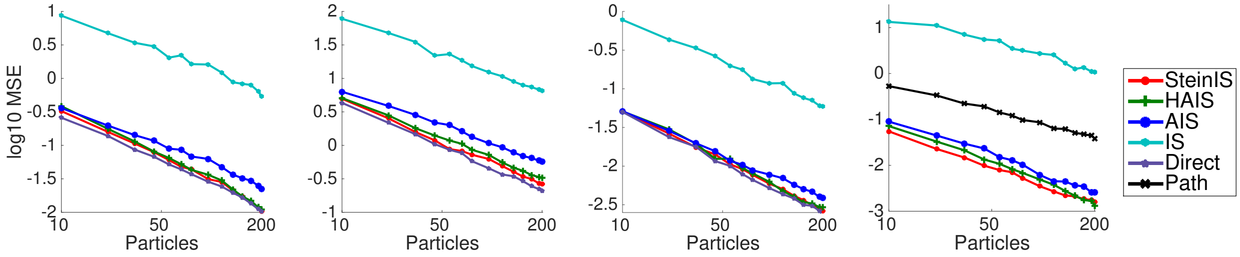
Figure 2: GMM with 10 mixture components. $d = 1$. In SVGD, 500 particles are evolved. In SteinIS, $|A| = 200$ and $|B| = 500$. For SVGD and SteinIS, all particles are drawn from the same Gaussian distribution $q_0(\boldsymbol{x})$.

We summarize some hyperparameters used in our experiments. We use RBF kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2/h)$, where h is the bandwidth. In most experiments, we let h=med$^2/(2\log(|A|+1))$, where med is the median of the pairwise distance of the current leader particles $\boldsymbol{x}_A^\ell$, and $|A|$ is the number of leader particles. The step sizes in our algorithms are chosen to be $\epsilon = \alpha/(1+\ell)^\beta$, where α and β are hyperparameters chosen from a validation set to achieve best performance. When $\epsilon \leq 0.1$, we use first-order approximation to calculate the determinants of Jacobian matrices as illustrated in proposition 1.

In what follows, we use "AIS" to refer to the annealing importance sampling with Langevin dynamics as its Markov transitions, and use "HAIS" to denote the annealing importance sampling whose Markov transition is Hamiltonian Monte Carlo (HMC). We use "transitions" to denote the number of intermediate distributions constructed in the paths of both SteinIS and AIS. A transition of HAIS may include L leapfrog steps, as implemented by Wu et al. (2016).

5.1 Gaussian Mixtures Models

We start with testing our methods on simple 2 dimensional Gaussian mixture models (GMM) with 10 randomly generated mixture components. First, we numerically investigate the convergence of KL divergence between the particle distribution q_t (in continuous time) and p. Sufficient particles are drawn and infinitesimal step ϵ is taken to closely simulate the continuous time system, as defined by (14),



(a) $\mathbb{E}[x]$ (b) $\mathbb{E}[x^2]$ (c) $\mathbb{E}[\cos(wx+b)]$ (d) Partition Function

Figure 3: 2D GMM with 10 randomly generated mixture components. (a)-(c) shows mean square error(MSE) for estimating $\mathbb{E}_p[h(x)]$, where $h(\boldsymbol{x}) = x_j,\ x_j^2,\ \cos(wx_j + b)$ with $w \sim \mathcal{N}(0, 1)$ and $b \in \text{Uniform}([0, 1])$ for $j = 1, 2$, and the normalization constant (which is 1 in this case). We used 800 transitions in SteinIS, HAIS and AIS, and take $L = 1$ in HAIS. We fixed the size of the leader particles $|A|$ to be 100 and vary the size of follower particles $|B|$ in SteinIS. The initial proposal q_0 is the standard Gaussian. "Direct" means that samples are directly drawn from p and is not applicable in (d). "IS" means we directly draw samples from q_0 and apply standard importance sampling. "Path" denotes path integration method in Algorithm 2 and is only applicable to estimate the partition function in (d). The MSE is averaged on each coordinate over 500 independent experiments for SteinIS, HAIS, AIS and Direct, and over 2000 independent experiments for IS. SVGD has similar results (not shown for clarity) as our SteinIS on (a), (b), (c), but can not be applied to estimate the partition function in task (d). The logarithm base is 10.

(15) and (16). Figrue 2(a)-(b) show that the KL divergence $\text{KL}(q_t, p)$, as well as the squared Stein discrepancy $\mathbb{D}(q_t, p)^2$, seem to decay exponentially in both SteinIS and the original SVGD. This suggests that the quality of our importance proposal q_t improves quickly as we apply sufficient transformations. However, it is still an open question to establish the exponential decay theoretically; see Liu (2017) for a related discussion.

We also empirically verify the convergence property of our SteinIS as the follower particle size $|B|$ increases (as the leader particle size $|A|$ is fixed). We apply SteinIS to estimate $\mathbb{E}_p[h(x)]$, where $h(\boldsymbol{x}) = x_j,\ x_j^2\ or\ \cos(wx_j + b)$ with $w \sim \mathcal{N}(0, 1)$ and $b \sim \text{Uniform}([0, 1])$ for $j = 1, 2$, and the partition function (which is trivially 1 in this case). From Figure 3, we can see that the mean square error(MSE) of our algorithms follow the typical convergence rate of IS, which is $O(1/\sqrt{|B|})$, where $|B|$ is the number of samples for performing IS. Figure 3 indicates that SteinIS can achieve almost the same performance as the exact Monte Carlo (which directly draws samples from the target p), indicating the proposal q_ℓ closely matches the target p.

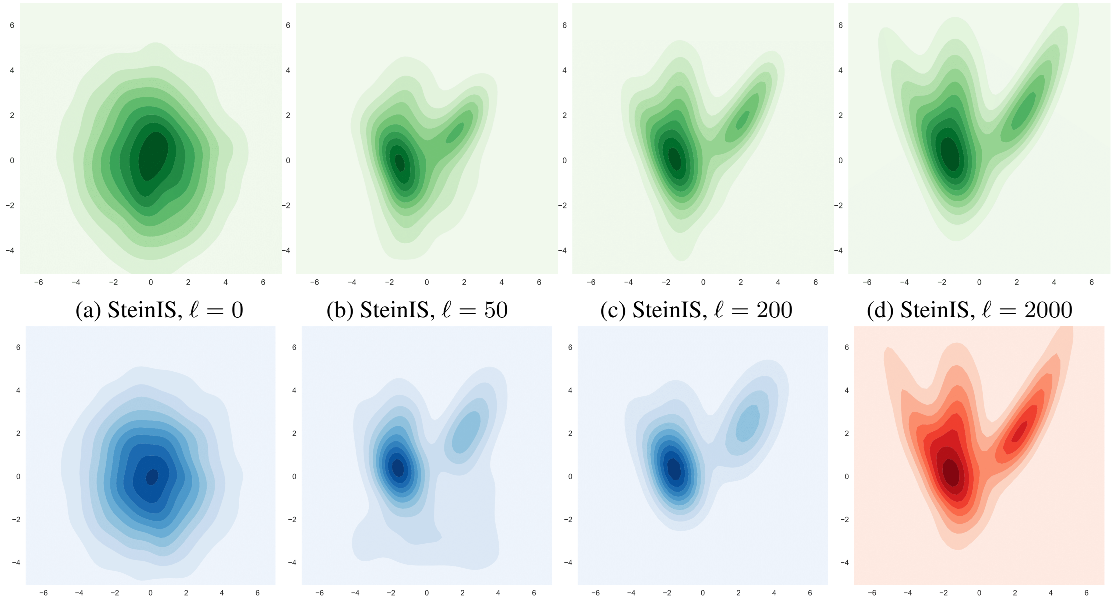
5.2 Comparison between SteinIS and Adaptive IS

In the following, we compare SteinIS with traditional adaptive IS (Ryu & Boyd, 2014) on a probability model $p(\boldsymbol{x})$, obtained by applying nonlinear transform on a three-component Gaussian mixture model. Specifically, let $\widetilde{q}$ be a 2D Gaussian mixture model, and $\boldsymbol{T}$ is a nonlinear transform defined by $\boldsymbol{T}(\boldsymbol{z}) = [a_1 z_1 + b_1, a_2 z_1^2 + a_3 z_2 + b_2]^\top$, where $\boldsymbol{z} = [z_1, z_2]^\top$. We define the target p to be the distribution of $\boldsymbol{x} = \boldsymbol{T}(\boldsymbol{z})$ when $\boldsymbol{z} \sim \widetilde{q}$. The contour of the target density p we constructed is shown in Figure 4(h). We test our SteinIS with $|A| = 100$ particles and visualize in Figure 4(a)-(d) the density of the evolved distribution q_ℓ using kernel density estimation, by drawing a large number of follower particles. We compare our method with the adaptive IS by (Ryu & Boyd, 2014) using a proposal family formed by Gaussian mixture with 200 components. The densities of the proposals obtained by adaptive IS at different iterations are shown in Figure 4(e)-(g). We can see that the evolved proposals of SteinIS converge to the target density $p(\boldsymbol{x})$ and approximately match $p(\boldsymbol{x})$ at 2000 iterations, but the optimal proposal of adaptive IS with 200 mixture components (at the convergence) can not fit $p(\boldsymbol{x})$ well, as indicated by Figure 4(g). This is because the Gaussian mixture proposal family (even with upto 200 components) can not closely approximate the non-Gaussian target distribution we constructed. We should remark that SteinIS can be applied to refine the optimal proposal given by adaptive IS to get better importance proposal by implementing a set of successive transforms on the given IS proposal.

Qualitatively, we find that the KL divergence (calculated via kernel density estimation) between our evolved proposal q_ℓ and p decreases to ≤ 0.003 after 2000 iterations, while the KL divergence between the optimal adaptive IS proposal and the target p can be only decreased to 0.42 even after sufficient optimization.

5.3 Gauss-Bernoulli Restricted Boltzmann Machine

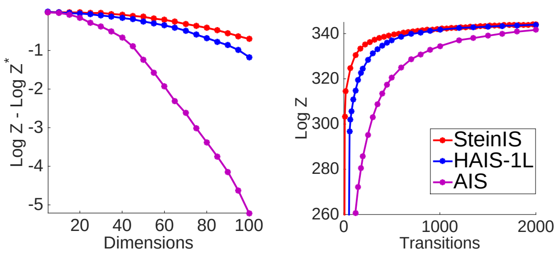
We apply our method to estimate the partition function of Gauss-Bernoulli Restricted Boltzmann Machine (RBM), which is a multi-modal, hidden variable graphical model. It consists of a continuous observable variable $\boldsymbol{x} \in \mathbb{R}^d$ and a binary hidden variable $\boldsymbol{h} \in \{\pm 1\}^{d'}$, with a joint probability



(a) SteinIS, $\ell = 0$ (b) SteinIS, $\ell = 50$ (c) SteinIS, $\ell = 200$ (d) SteinIS, $\ell = 2000$

(e) Adap IS, 0 iteration (f) Adap IS, 1000 iteration (g) Adap IS, 10000 iteration (h) Exact

Figure 4: Evolution of the contour of density functions for SteinIS and Adaptive IS. The top row (a)-(d) shows the contours of the evolved density functions in SteinIS, and bottom row (c)-(g) are the evolved contours of the traditional adaptive IS with Gaussian mixture proposals. (h) is the contour of the target density p. The number of the mixture components for adaptive IS is 200 and the number of leader particles for approximating the map in SteinIS is also 200.



(a) Vary dimensions (b) 100 dimensions

Figure 5: Gauss-Bernoulli RBM with $d' = 10$ hidden variables. The initial distribution $q_0(\boldsymbol{x})$ for all the methods is a same multivariate Gaussian. We let $|A| = 100$ in SteinIS and use $(B\ =)100$ importance samples in SteinIS, HAIS and AIS. In (a), we use 1500 transitions for HAIS, SteinIS and AIS. "HAIS-1L" means we use $L = 1$ leapfrog in each Markov transition of HAIS. $\log Z^*$ denotes the logarithm of the exact normalizing constant. All experiments are averaged over 500 independent trails.

density function of form

$$p(\boldsymbol{x}, \boldsymbol{h}) = \frac{1}{Z}\exp(\boldsymbol{x}^{\mathrm{T}}B\boldsymbol{h} + b^{\mathrm{T}}\boldsymbol{x} + c^{\mathrm{T}}\boldsymbol{h} - \frac{1}{2}\|\boldsymbol{x}\|_2^2), \quad (21)$$

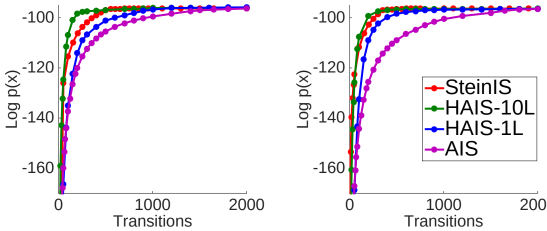
where $p(\boldsymbol{x}) = \frac{1}{Z}\sum_{\boldsymbol{h}} p(\boldsymbol{x}, \boldsymbol{h})$ and Z is the normalization constant. By marginalzing the hidden variable h, we can show that $p(\boldsymbol{x})$ is

$$p(\boldsymbol{x}) = \frac{1}{Z}\exp(b^{\mathrm{T}}\boldsymbol{x} - \frac{1}{2}\|\boldsymbol{x}\|_2^2)\prod_{i=1}^{d'}[\exp(\varphi_i) + \exp(-\varphi_i)],$$

where $\varphi = B^{\mathrm{T}}\boldsymbol{x} + c$, and its score function $\boldsymbol{s}_p$ is easily derived as

$$\boldsymbol{s}_p(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}) = b - \boldsymbol{x} + B\frac{\exp(2\varphi) - 1}{\exp(2\varphi) + 1}.$$

In our experiments, we simulate a true model $p(\boldsymbol{x})$ by drawing b and c from standard Gaussian and select B uniformly random from $\{0.5, -0.5\}$ with probability 0.5. The dimension of the latent variable $\boldsymbol{h}$ is 10 so that the probability model $p(\boldsymbol{x})$ is the mixture of 2^{10} multivariate Gaussian distribution. The exact normalization constant Z can be feasibly calculated using the brute-force algorithm in this case. Figure 5(a) and Figure 5(b) shows the performance of SteinIS on Gauss-Bernoulli RBM when we vary the dimensions of the observed variables and the number of transitions in SteinIS, respectively. We can see that SteinIS converges slightly faster than HAIS which uses one leapfrog step in each of its Markov transition. Even with the same number of Markov transitions, AIS with Langevin dynamics converges much slower than both SteinIS and HAIS. The better performance of HAIS comparing to AIS was also observed by Sohl-Dickstein & Culpepper (2012) when they first proposed HAIS.



(a) 20 hidden variables (b) 50 hidden variables

Figure 6: Calculating the testing log-likelihood $\log p(x)$ for the deep generative model on MNIST. The initial distribution q_0 used in SteinIS, HAIS and AIS is a same multivariate Gaussian. We let $|A| = 60$ in SteinIS and use 60 samples for each image to implement IS in HAIS and AIS. "HAIS-10L" and "HAIS-1L" denote using $L = 10$ and $L = 1$ in each Markov transition of HAIS, respectively. The log-likelihood $\log p(x)$ is averaged over 1000 images randomly chosen from MNIST. Figure (a) and (b) show the results when using 20 and 50 hidden variables, respectively. Note that the dimension of the observable variable $\boldsymbol{x}$ is fixed, and is the size of the MNIS images.

5.4 Deep Generative Models

Finally, we implement our SteinIS to evaluate the log-likelihoods of the decoder models in variational autoencoder (VAE) (Kingma & Welling, 2013). VAE is a directed probabilistic graphical model. The decoder-based generative model is defined by a joint distribution over a set of latent random variables $\boldsymbol{z}$ and the observed variables $\boldsymbol{x}$: $p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{z})$. We use the same network structure as that in Kingma & Welling (2013). The prior $p(\boldsymbol{z})$ is chosen to be a multivariate Gaussian distribution. The log-likelihood is defined as $p(\boldsymbol{x}) = \int p(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$, where $p(\boldsymbol{x} \mid \boldsymbol{z})$ is the Bernoulli MLP as the decoder model given in Kingma & Welling (2013). In our experiment, we use a two-layer network for $p(\boldsymbol{x} \mid \boldsymbol{z})$, whose parameters are estimated using a standard VAE based on the MNIST training set. For a given observed test image $\boldsymbol{x}$, we use our method to sample the posterior distribution $p(\boldsymbol{z}|\boldsymbol{x}) = \frac{1}{p(\boldsymbol{x})}p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$, and estimate the partition function $p(x)$, which is the testing likelihood of image $\boldsymbol{x}$.

Figure 6 also indicates that our SteinIS converges slightly faster than HAIS-1L which uses one leapfrog step in each of its Markov transitions, denoted by HAIS-1L. Meanwhile, the running time of SteinIS and HAIS-1L is also comparable as provided by Table 1. Although HAIS-10L, which use 10 leapfrog steps in each of its Markov transition, converges faster than our SteinIS, it takes much more time than our SteinIS in our implementation since the leapfrog steps in the Markov transitions of HAIS are sequential and can not be parallelized. See Table 5.4. Compared with HAIS and AIS, our SteinIS has another advantage: if we want to increase the transitions from 1000 to 2000 for better accuracy, SteinIS can build on the result from 1000 transitions and just need to run another 1000 iterations, while HAIS cannot take advantage of the result from 1000 transitions and have to independently run another 2000 transitions.

Table 1: Running Time (in seconds) on MNIST, using the same setting as in Figure 6. We use 1000 transitions in all methods to test the running time.

Dimensions of $\boldsymbol{z}$	10	20	50
SteinIS	224.40	226.17	261.76
HAIS-10L	600.15	691.86	755.44
HAIS-1L	157.76	223.30	256.23
AIS	146.75	206.89	230.14

6 CONCLUSIONS

In this paper, we propose an nonparametric adaptive importance sampling algorithm which leverages the nonparametric transforms of SVGD to maximumly decrease the KL divergence between our importance proposals and the target distribution. Our algorithm turns SVGD into a typical adaptive IS for more general inference tasks. Numerical experiments demonstrate that our SteinIS works efficiently on the applications such as estimating the partition functions of graphical models and evaluating the log-likelihood of deep generative models. Future research includes improving the computational and statistical efficiency in high dimensional cases, more theoretical investigation on the convergence of $\mathrm{KL}(q_\ell \,||\, p)$, and incorporating Hamiltonian Monte Carlo into our SteinIS to derive more efficient algorithms.

Acknowledgments
This work is supported in part by NSF CRII 1565796. We thank Yuhuai Wu from Toronto for his valuable comments.

References

Berlinet, Alain and Thomas-Agnan, Christine. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

Braun, W and Hepp, K. The Vlasov dynamics and its fluctuations in the 1/n limit of interacting classical particles. *Communications in mathematical physics*, 56(2): 101–113, 1977.

Cappé, Olivier, Douc, Randal, Guillin, Arnaud, Marin, Jean-Michel, and Robert, Christian P. Adaptive importance sampling in general mixture classes. *Statistics and Computing*, 18(4):447–459, 2008.

Chwialkowski, Kacper, Strathmann, Heiko, and Gretton, Arthur. A kernel test of goodness of fit. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

Cotter, Colin, Cotter, Simon, and Russell, Paul. Parallel adaptive importance sampling. *arXiv preprint arXiv:1508.01132*, 2015.

Del Moral, Pierre. *Mean field simulation for Monte Carlo integration*. CRC Press, 2013.

Gelman, Andrew and Meng, Xiao-Li. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical science*, pp. 163–185, 1998.

Gorham, Jackson and Mackey, Lester. Measuring sample quality with Stein's method. In *Advances in Neural Information Processing Systems*, pp. 226–234, 2015.

Gorham, Jackson and Mackey, Lester. Measuring sample quality with kernels. *arXiv preprint arXiv:1703.01717*, 2017.

Gretton, Arthur, Fukumizu, Kenji, Harchaoui, Zaid, and Sriperumbudur, Bharath K. A fast, consistent kernel two-sample test. In *Advances in neural information processing systems*, pp. 673–681, 2009.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hinton, Geoffrey E and Salakhutdinov, Ruslan R. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

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

Jourdain, B and Méléard, S. Propagation of chaos and fluctuations for a moderate model with smooth initial data. In *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, volume 34, pp. 727–766. Elsevier, 1998.

Kingma, Diederik P and Welling, Max. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kingma, Diederik P, Salimans, Tim, Jozefowicz, Rafal, Chen, Xi, Sutskever, Ilya, and Welling, Max. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pp. 4743–4751, 2016.

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

Liu, Qiang and Wang, Dilin. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pp. 2370–2378, 2016.

Liu, Qiang, Lee, Jason D, and Jordan, Michael I. A kernelized Stein discrepancy for goodness-of-fit tests. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

Marzouk, Youssef, Moselhy, Tarek, Parno, Matthew, and Spantini, Alessio. An introduction to sampling via measure transport. *arXiv preprint arXiv:1602.05023*, 2016.

Neal, Radford M. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

Oates, Chris J, Girolami, Mark, and Chopin, Nicolas. Control functionals for Monte Carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2016.

Rezende, Danilo and Mohamed, Shakir. Variational inference with normalizing flows. In *Proceedings of The 32nd International Conference on Machine Learning*, pp. 1530–1538, 2015.

Ryu, Ernest K and Boyd, Stephen P. Adaptive importance sampling via stochastic convex programming. *arXiv preprint arXiv:1412.4845*, 2014.

Sohl-Dickstein, Jascha and Culpepper, Benjamin J. Hamiltonian annealed importance sampling for partition function estimation. *arXiv preprint arXiv:1205.1925*, 2012.

Spantini, Alessio, Bigoni, Daniele, and Marzouk, Youssef. Inference via low-dimensional couplings. *arXiv preprint arXiv:1703.06131*, 2017.

Spohn, Herbert. *Large scale dynamics of interacting particles*. Springer Science & Business Media, 2012.

Sznitman, Alain-Sol. Topics in propagation of chaos. In *Ecole d'été de probabilités de Saint-Flour XIX1989*, pp. 165–251. Springer, 1991.

Welling, Max, Rosen-Zvi, Michal, and Hinton, Geoffrey E. Exponential family harmoniums with an application to information retrieval. In *Nips*, volume 4, pp. 1481–1488, 2004.

Wu, Yuhuai, Burda, Yuri, Salakhutdinov, Ruslan, and Grosse, Roger. On the quantitative analysis of decoder-based generative models. *arXiv preprint arXiv:1611.04273*, 2016.