Stein Variational Gradient Descent: A Survey

Stellenbosch
UNIVERSITY
IYUNIVESITHI
UNIVERSITEIT

Leon Halgryn

Assignment presented in the partial fulfilment
of the requirement for the degree of
Master of Science (MSc) Machine Learning and Artificial Intelligence
at Stellenbosch University

Supervisor: Prof. S. Kroon

November 2023

PLAGIARISM DECLARATION

1. Plagiarism is the use of ideas, material and other intellectual property of another's work and to present it as my own.
2. I agree that plagiarism is a punishable offence because it constitutes theft.
3. Accordingly, all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism.
4. I also understand that direct translations are plagiarism.
5. I declare that the work contained in this assignment, except otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this assignment or another assignment.

21946345	
Student number	Signature
L.P. Halgryn	06 November 2023
Initials and surname	Date

ACKNOWLEDGEMENTS

Firstly, I want to thank my parents, Leon and Tessa Halgryn, and my siblings, Andre and Rita Halgryn, for their unwavering support and encouragement throughout the course of my studies. Secondly, I want to thank my supervisor, Prof. Steve Kroon, for his invaluable insights and guidance on this project. Lastly, I want to thank our masters programme coordinator, Prof. Willie Brink.

ABSTRACT

In the realm of Bayesian machine learning and probabilistic modelling, practitioners are often faced with complex/intractable posterior distributions that cannot be evaluated in closed form, and hence are not amenable to use in downstream tasks. Hence, accurate approximation of such distributions remains an important and challenging problem.

Stein variational gradient descent (SVGD) has recently been proposed to address this problem by iteratively applying deterministic updates to a set of particles to match the posterior distribution of interest. To this end, the particle positions are iteratively updated by following the direction of steepest descent on reverse KL divergence within a ball of reproducing kernel Hilbert space. The development of SVGD stems from a discrepancy measure used in goodness-of-fit tests that has roots in Stein's method in theoretical statistics.

This study provides a comprehensive survey of SVGD and its theoretical underpinnings, aiming to shed light on the algorithm and its applicability in practice. Furthermore, this study discusses an implementation of SVGD in reinforcement learning known as the Stein variational policy gradient (SVPG) method. A novel variant of SVPG is introduced that leverages ideas from existing variants of SVGD. Several experiments are conducted to demonstrate the performance of SVGD and SVPG.

Our results indicate that SVGD shows significant promise for approximating complex target distributions, and outperforms several well-known MCMC algorithms on a simple sampling experiment. Furthermore, the results demonstrate that SVPG and our variant thereof outperform the REINFORCE policy gradient method on classic control and Box2D gym problems.

Key words:
Variational inference; Stein's method; particle-based optimisation; kernel-based methods; reinforcement learning; policy gradient methods.

LIST OF APPENDICES

LIST OF ABBREVIATIONS AND/OR ACRONYMS

SVGD	Stein Variational Gradient Descent
SVPG	Stein Variational Policy Gradient
KSD	Kernelised Stein discrepancy
GOF	Goodness-of-fit
IPM	Integral probability metric
i.i.d	Independent and identically distributed
RKHS	Reproducing kernel Hilbert space
KL	Kullback-Leibler
VI	Variational inference
GMM	Gaussian mixture model
MH	Metropolis-Hastings
HMC	Hamiltonial/Hybrid Monte Carlo
NUTS	No U-Turn sampler
RBF	Radial basis function
BL	Bounded Lipschitz
MCMC	Markov chain Monte Carlo
MAP	Maximum *a posteriori*
GAN	Generative adversarial network
VAE	Variational autoencoder
BNN	Bayesian neural network
COD	Curse of dimensionality
RL	Reinforcement learning

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In the realm of probabilistic modelling and Bayesian machine learning, accurate approximation of (or sampling from) high-dimensional target distributions poses a significant challenge, especially in the context of Bayesian inference for complex posterior distributions. Stein Variational Gradient Descent (SVGD), proposed by Liu & Wang (2016), is a general-purpose, deterministic particle-based variational inference algorithm that attempts to address this challenge by incrementally transforming a simple base distribution (e.g. the standard normal distribution) to approximate the posterior distribution of interest. This is achieved by iteratively transporting a set of sample points (referred to as particles) from the base distribution to approximate the posterior distribution. SVGD leverages gradient information of the target distribution to guide particles toward high-density regions of the posterior, and utilises kernel information to simultaneously allow sharing of information across particles and to provide a deterministic repulsive force that prevents particles from collapsing onto a single mode.

The purpose of this study is to provide a comprehensive survey of SVGD together with its theoretical underpinnings, aiming to shed light on the applicability of SVGD in practice. We hope to make SVGD more accessible to machine learning practitioners who may have a limited background in statistics.

While the report is written to be mostly self-contained, we assume that readers are familiar with basic concepts from statistics and measure theory, and are familiar with concepts in functional analysis, particularly reproducing kernel Hilbert spaces.

1.2 OUTLINE OF THIS PAPER

This report is organised as follows. Chapter 2 presents a comprehensive development of SVGD, including its theoretical underpinnings in Stein's method, Stein discrepancy, and kernelised Stein discrepancy. Furthermore, a simple illustration of SVGD is given in the context of sampling, and the convergence properties of SVGD are discussed. Due to the statistical nature and background of SVGD, Chapter 2 is more mathematical than later chapters, and is presented in the form of definitions and theorems, whereas later chapters are more focused on intuitive discussions. Chapter 3 discusses the major advantages and limitations of SVGD, together with a comparison of SVGD to variational inference and Markov chain Monte Carlo methods. Furthermore, several variants of SVGD are discussed that may alleviate some of its major limitations. Chapter 4 discusses an implementation of SVGD in the context of reinforcement learning known as the Stein Variational Policy Gradient (Liu *et al.*, 2017) (SVPG) method, which is used for training a set of diverse policies. Furthermore, a novel variant of SVPG is presented and several experiments are conducted. The report is concluded in Chapter 5.

1.3 NOTATION AND TERMINOLOGY

To simplify notation and maintain generality, we do not distinguish between a measure, P, and a distribution P. We sometimes refer to a distribution P by its density function, p, as the density function uniquely characterizes the distribution. Throughout this paper, multivariate distributions and functions are assumed, with univariate cases presented in footnotes where applicable. The theory presented applies to both univariate and multivariate settings, with minimal adjustments. Hence, our notation does not explicitly distinguish univariate from multivariate functions and distributions.

1.4 NOTE ON GITHUB REPOSITORY

There is a GitHub repository accompanying this report available at `https://github.com/lhalgryn/Stein-Variational-Gradient-Descent` containing the following:

1. Background information on measure theory and reproducing kernel Hilbert spaces in the form of Jupyter Notebooks.
2. Code to reproduce the experiments and results contained in this report.
3. A Jupyter Notebook assignment on SVGD.

Furthermore, the repository contains a README.txt file that explains how the code can be used to reproduce the results presented in this report.

CHAPTER 2

STATISTICAL DEVELOPMENT OF STEIN VARIATIONAL GRADIENT DESCENT

2.1 INTRODUCTION

Stein Variational Gradient Descent (SVGD) (Liu *&* Wang, 2016) is a variational inference algorithm with roots in Stein's method, which is an approach used in theoretical statistics to bound a given measure of distance between two distributions. Traditionally, Stein's method has primarily been used as a tool for proving central limit theorems. However, the advent of kernelised Stein discrepancy (KSD) (Liu *et al.*, 2016; Chwialkowski *et al.*, 2016) has given rise to a powerful framework for conducting goodness-of-fit (GOF) tests, and subsequently to a flexible framework for variational inference.

This chapter is dedicated to the statistical development SVGD. The chapter aims to provide a comprehensive understanding of the statistical foundations of SVGD, including its theoretical underpinnings and practical implementation. This chapter begins with relevant background information in Section 2.2. In Section 2.3, an overview of Stein's method is provided. Section 2.4 discusses the Stein discrepancy and its kernelised variant is presented in Section 2.5. Section 2.6 delves into the SVGD algorithm and illustrates the usage of SVGD on a basic sampling experiment. Finally, Section 2.7 provides an overview of the convergence properties of SVGD and a conclusion is given in Section 2.8.

2.2 BACKGROUND

Before we discuss Stein's method in general, we first discuss Stein's characterisation of the normal distribution (Stein, 1972), which has since become known as *Stein's lemma*. Furthermore, we provide a brief overview of integral probability metrics (IPMs), for which bounds can be obtained via Stein's method.

2.2.1 Stein's Lemma

Lemma 2.1. (Stein's lemma)

Let $Z \sim \mathcal{N}(0,1)$, and let $f : \mathbb{R} \to \mathbb{R}$ be an absolutely continuous function such that $\mathbb{E}\,|f'(Z)| < \infty$ (Chatterjee, 2014). Then the following holds:

$$\mathbb{E}\left[Zf(Z)\right] = \mathbb{E}\left[f'(Z)\right] \quad .$$

Proof. See Appendix A. □

Lemma 2.1 characterises the standard normal distribution in the sense that the lemma holds for no other distribution of Z (Stein, 1972).

Corollary 2.2.

While Stein's lemma (Lemma 2.1) is given for the univariate standard normal case, it can be generalised to both the non-standard and multivariate cases: if $X \sim \mathcal{N}(\mu, \sigma^2)$ and $f : \mathbb{R} \to \mathbb{R}$ is absolutely continuous, then

$$\frac{1}{\sigma^2}\mathbb{E}\left[(X-\mu)f(X)\right] = \mathbb{E}\left[f'(X)\right] \quad .$$

Proof. See Appendix A. □

Remark: The converse of Stein's lemma is also true: if $\frac{1}{\sigma^2}\mathbb{E}\left[(X-\mu)f(X)\right] = \mathbb{E}\left[f'(X)\right]$, then $X \sim \mathcal{N}(\mu, \sigma^2)$.

The remarkable property of Stein's lemma is that it enables conclusions to be drawn about the *approximate* normality of arbitrary random variables X. Suppose $X \sim Q$ is an arbitrary, real-valued random variable with mean μ and variance σ^2, and suppose that the relationship,

$$\frac{1}{\sigma^2}\mathbb{E}_{X\sim Q}\left[(X-\mu)f(X)\right] \approx \mathbb{E}_{X\sim Q}\left[f'(X)\right]$$

holds for all continuous functions f in a rich class of functions $\mathcal{F}$, then we may conclude that X is approximately normally distributed, i.e. $X \dot{\sim} \mathcal{N}(\mu, \sigma^2)$.

Stein's method, presented in Section 2.3, generalises and formalises this line of reasoning to distributions beyond the normal, and allows comparing the distribution Q to arbitrary target distributions.

2.2.2 Integral probability metrics

Integral probability metrics (IPMs), defined below, are a special class of probability metrics that can be used to quantify the distance between distributions (e.g., Müller, 1997).

Definition 2.3. (Integral Probability Metric)
Let P and Q be probability measures defined on a measurable space $(\Omega, \mathcal{X})$, and let $\mathcal{H}$ denote a class of bounded, real-valued measurable functions on $\mathcal{X}$. An IPM on $\mathcal{X}$ for measuring the distance between the distributions P and Q takes the following form as given by Sriperumbudur et al. (2009):

$$\begin{aligned} d_{\mathcal{H}}(Q,P) :&= \sup_{h \in \mathcal{H}} \left| \int_{\mathcal{X}} h dQ - \int_{\mathcal{X}} h dP \right| && (2.1) \\ &= \sup_{h \in \mathcal{H}} \left| \mathbb{E}_{X \sim Q}\left[h(X)\right] - \mathbb{E}_{Y \sim P}\left[h(Y)\right] \right| \ . && (2.2) \end{aligned}$$

Remark: The class of functions $\mathcal{H}$ is a "measure-determining class of test functions" (Anastasiou *et al.*, 2023) that induces/specifies the specific IPM. Section 2.4 discusses the *Stein discrepancy* measure as a special case of an IPM (Gong *et al.*, 2021) in which members of the function class $\mathcal{H}$ are chosen to yield zero expectation under the target distribution P (Hu *et al.*, 2021; Gorham *et al.*, 2020). This is especially useful in settings where the target distribution is complex and cannot be evaluated in closed form, and hence expectations under the target distribution are intractable to compute.

Remark: Suppose we are given a sequence of probability measures $\{Q_n\}$ for approximating an arbitrary target probability measure P. Provided the class of functions $\mathcal{H}$ in Definition 2.3 is sufficiently rich, the IPM induced by $\mathcal{H}$ metricises weak convergence (Gorham *&* Mackey, 2015). This means that $d_{\mathcal{H}}(Q_n, P) \to 0$ as $n \to \infty \implies Q_n$ weakly converges to the target P, which we denote by $Q_n \rightharpoonup P$.

2.3 STEIN'S METHOD

For the remainder of this section, we assume that we are working with two probability measures, P and Q, defined on a measurable space $(\Omega, \mathcal{X})$, where $\mathcal{X} \subseteq \mathbb{R}^d$, with continuous probability density functions p and q respectively. Stein's method is a three-step procedure for bounding the distance (in the form of an IPM) between the two distributions (Gorham *et al.*, 2019). Given a reference IPM in the form of Equation (2.2), Stein's method can be used to obtain (upper, lower, or two-sided) bounds on the distance $d_{\mathcal{H}}(Q, P)$.

We now discuss each of the three steps in Stein's method in turn. See, Ley *et al.* (2014) and Ross (2011), for example, for more detailed discussions on Stein's method.

Step 1: Stein Operator

The first step in Stein's method involves specifying/constructing a so-called *Stein operator* to characterise the distribution of interest. See Anastasiou *et al.* (2023), for example, for a discussion of several well-known methods to construct Stein operators.

Definition 2.4. (Stein operator)
Given a distribution P with probability density function p, and a class of functions $\mathcal{F}$, a Stein operator $\mathcal{T}_p : \mathcal{F} \to \mathbb{R}^d$ for characterising P, is an operator acting on functions $f \in \mathcal{F}$ such that the following holds:

$$\mathbb{E}_{X \sim P}\left[\mathcal{T}_p f(X)\right] = 0 \; \forall f \in \mathcal{F} \iff X \sim P \ . \tag{2.3}$$

*The class of functions $\mathcal{F}$ for which the above holds is called a **Stein class** for the distribution P, which we denote by $\mathcal{F}(\mathcal{T}_p)$, and the equivalence in Equation (2.3) is called a Stein characterisation (Anastasiou et al., 2023).*

Example 2.5. (Stein operator for the standard normal distribution)
Recall that Stein's lemma (Lemma 2.1) for the standard normal distribution states that $\mathbb{E}\left[Zf(Z)\right] = \mathbb{E}\left[f'(Z)\right]$ for all absolutely continuous functions f if and only if $Z \sim \mathcal{N}(0,1)$. This implies that:

$$\mathbb{E}\left[f'(Z) - Zf(Z)\right] = 0 \; \forall f \in \mathcal{C}^1(\mathbb{R}) \iff Z \sim \mathcal{N}(0,1) \ .$$

This gives rise to the following Stein operator for characterising the standard normal distribution with probability density function $\phi(z)$:

$$\mathcal{T}_\phi f(z) = f'(z) - zf(z)$$

with the corresponding Stein class given by $\mathcal{F}(\mathcal{T}_\phi) = \mathcal{C}^1(\mathbb{R})$, the space of continuous, real-valued functions with a continuous first derivative. In Section 2.4, we will show that the Stein operator *for the standard normal distribution is an example of a* ***Langevin-Stein*** *operator.*

Step 2: Stein Equation

Given a Stein operator (and corresponding Stein class) for the target distribution P, the second step in Stein's method involves the so-called *Stein equation*, defined below.

Definition 2.6. (Stein equation)
Given a Stein operator $\mathcal{T}_p$ and corresponding Stein class $\mathcal{F}(\mathcal{T}_p)$, the Stein equation is given by Gaunt et al. (2019) as:

$$\mathcal{T}_p f_h(x) = h(x) - \mathbb{E}_{Y\sim P}\left[h(Y)\right] \tag{2.4}$$

where $h \in \mathcal{H}$ is a bounded test function and $f_h \in \mathcal{F}(\mathcal{T}_p)$ is the solution we seek. Here, $\mathcal{H}$ is a class of test functions as in Definition 2.3.

The second step in Stein's method requires proving that a solution $f_h \in \mathcal{F}(\mathcal{T}_p)$ exists for every (bounded) test function $h \in \mathcal{H}$ (Gorham *et al.*, 2019).

Remark: Bounding the $\mathcal{H}$-dependent IPM using Stein's method requires that a solution to Stein's equation exists for every $h \in \mathcal{H}$. Though a derivation is beyond the scope of this report, Ley *et al.* (2017) and Mijoule *et al.* (2023) show that, if the Stein operator $\mathcal{T}_p$ is chosen appropriately, then a well-defined solution $f_h \in \mathcal{F}(\mathcal{T}_p)$ to Stein's equation exists for every bounded, continuous test function $h \in \mathcal{H}$.

Example 2.7. (Stein equation for the standard normal distribution)
Suppose we are given an arbitrary random variable $X \sim Q$ and we want to determine if X is (approximately) standard normally distributed, i.e. $X \sim \mathcal{N}(0,1)$. Recall from Example 2.5 that the Stein operator characterising the standard normal distribution is given by $\mathcal{T}_\phi f(x) = f'(x) - xf(x)$. Hence, the Stein equation may be given by:

$$f_h'(x) - xf_h(x) = h(x) - \mathbb{E}_{Y\sim\mathcal{N}(0,1)}\left[h(Y)\right] \quad .$$

Replacing the quantity x with the random variable X and taking expectations with respect to $X \sim Q$ yields:

$$\mathbb{E}_{X\sim Q}\left[f_h'(X) - Xf_h(X)\right] = \mathbb{E}_{X\sim Q}\left[h(X)\right] - \mathbb{E}_{Y\sim\mathcal{N}(0,1)}\left[h(Y)\right] \quad .$$

If we assume that $Q = \mathcal{N}(0,1)$, and that X and Y are i.i.d, we arrive at:

$$\mathbb{E}_{X\sim\mathcal{N}(0,1)}\left[f_h'(X) - Xf_h(X)\right] = 0 \ ,$$

which is exactly the result given in Stein's lemma (Lemma 2.1).

Example 2.7 demonstrates how Stein's method can be used to compare an arbitrary distribution Q to the standard normal target distribution $P = \mathcal{N}(0,1)$.

Step 3: Obtaining bounds

The final step in Stein's method involves bounding the reference IPM, $d_{\mathcal{H}}(Q,P)$. We now assume that a solution $f_h \in \mathcal{F}(\mathcal{T}_p)$ to Stein's equation - see Equation (2.4) - exists for every test function $h \in \mathcal{H}$. Let $\mathfrak{F} \subseteq \mathcal{F}(\mathcal{T}_p)$ denote the space of solutions to Stein's equation, i.e. $\mathfrak{F} = \{f_h \in \mathcal{F}(\mathcal{T}_p) | f_h \text{ is a solution to Stein's equation}\}$. If we replace the quantity x in Equation (2.4) with the random variable X and take expectations with respect to the distribution Q, we arrive at:

$$\mathbb{E}_{X\sim Q}\left[\mathcal{T}_p f_h(X)\right] = \mathbb{E}_{X\sim Q}\left[h(X)\right] - \mathbb{E}_{Y\sim P}\left[h(Y)\right] \ .$$

This then allows us to rewrite the IPM in Equation (2.2) as follows:

$$\begin{aligned} d_{\mathcal{H}}(Q,P) &= \sup_{h\in\mathcal{H}}\Big|\mathbb{E}_{X\sim Q}\left[h(X)\right] - \mathbb{E}_{Y\sim P}\left[h(Y)\right]\Big| \\ &= \sup_{f_h\in\mathfrak{F}}\left|\mathbb{E}_{X\sim Q}\left[\mathcal{T}_p f_h(X)\right]\right| \\ &\leq \sup_{f_h\in\mathcal{F}(\mathcal{T}_p)}\Big|\mathbb{E}_{X\sim Q}\left[\mathcal{T}_p f_h(X)\right]\Big| \ . \end{aligned} \tag{2.5}$$

If we further assume that the class of functions $\mathfrak{F}$ is closed under negation, i.e. $f \in \mathfrak{F} \implies -f \in \mathfrak{F}$, then we can ignore the absolute value above (this is also true for $\mathcal{H}$ and $\mathcal{F}(\mathcal{T}_p)$).

In the following section, we will show that the final quantity in Equation (2.5) gives rise to a so-called *Stein discrepancy*, which is lower-bounded by the reference IPM, $d_{\mathcal{H}}(Q,P)$.

The final step in Stein's method then involves lower-bounding the penultimate quanity in Equation (2.5), i.e. lower-bounding $\sup_{f_h\in\mathfrak{F}}\left|\mathbb{E}_{X\sim Q}\left[\mathcal{T}_p f_h(X)\right]\right|$. One possible approach for obtaining such bounds is the coupling technique by Reinert (1998). See Bonis (2020), for example, for bounding the 2-Wasserstein distance between an arbitrary distribution Q, and the multivariate normal distribution.

Remark: Stein's method is useful for bounding reference IPMs since it is generally easier to obtain bounds on the penultimate quantity in Equation (2.5) than for the IPM itself. This is, in part, due to the fact that the penultimate quantity in Equation (2.5) does not require explicit integration under the target distribution P (Gorham *et al.*, 2019). This is especially useful when P is a complex distribution that cannot be evaluated in closed form.

2.4 STEIN DISCREPANCY

In the previous section, we demonstrated how Stein's method facilitates distributional comparisons by simplifying the task of bounding the distance between distributions, where the measure of distance is given by a predetermined IPM. In this section, we show that Stein's method gives rise to a new measure of distance between two distributions that does not involve a reference IPM. Given a distribution Q that we wish to compare to an arbitrary target distribution P, both supported on a measurable space $(\Omega, \mathcal{X})$, where $\mathcal{X} \subseteq \mathbb{R}^d$, the so-called *Stein discrepancy* directly quantifies the similarity or dissimilarity between Q and P.

Before we discuss the Stein discrepancy measure, we first provide a motivation for discrepancy measures in the context of goodness-of-fit (GOF) tests.

2.4.1 Preliminaries

Goodness-of-fit tests In general, GOF tests refer to statistical procedures used to assess whether a given sample of data follows some hypothesised distribution: given an independent and identically distributed (i.i.d) sample of observed data $\{x_i\}_{i=1}^n$ from an unknown distribution Q with probability density/mass function $q(x)$, we would like to test whether the data could have plausibly arisen from some hypothesised/target distribution P with probability density/mass function $p(x)$. This is formally expressed by the following hypothesis test:

$$H_0 : p = q \quad \equiv \quad H_0 : \{x_i\}_{i=1}^n \overset{i.i.d}{\sim} p(x) \quad . \tag{2.6}$$

Traditional GOF testing procedures such as the $\mathcal{X}^2$ test are often limited in flexibility due to stringent assumptions on the distributions P and Q [1] (see, Rice, 2007:chap. 9). A more flexible approach to conduct GOF tests would be to consider some measure of similarity or dissimilarity between the distributions P and Q. This is where the notion of discrepancy measures comes into play. In general, a discrepancy measure $\mathbb{D}(q||p)$ is a non-negative functional that quantifies the similarity/dissimilarity between two distributions, P and Q, with the following property:

$$\mathbb{D}(q||p) = 0 \iff p = q \ . \tag{2.7}$$

Given a discrepancy measure $\mathbb{D}$, we can conduct the GOF test in Equation (2.6) by considering the deviation of $\mathbb{D}(q||p)$ from zero [2]. If we find that $\mathbb{D}(q||p)$ is significantly different from zero (at some significance level α), we reject the null hypothesis and conclude that $p \neq q$ [3].

2.4.2 Development of Stein Discrepancy

This section constructs a Stein discrepancy measure based on the so-called *Langevin-Stein operator* developed by Gorham & Mackey (2015) using the generator approach by Barbour (1990) as the infinitesimal generator of the Langevin diffusion (Gorham *et al.*, 2019).

Definition 2.8. (Langevin-Stein operator)
The Langevin-Stein operator for a distribution P with continuous density function p is given by:

$$\mathcal{T}_p f(x) = s_p(x) f(x)^T + \nabla_x f(x) \ , \tag{2.8}$$

where $s_p : \mathcal{X} \to \mathbb{R}^d$ is the Stein score function defined by $s_p(x) := \nabla_x \log p(x)$, and $f : \mathcal{X} \to \mathbb{R}^{d'}$ is a continuously differentiable vector-valued function of the form $f(x) = \begin{bmatrix} f_1(x) & f_2(x) & \dots & f_{d'}(x) \end{bmatrix}^T$. Here, ∇f and $\mathcal{T}_p f$ are $d \times d'$ vector-valued functions [4].

Example 2.9. *As pointed out in Example 2.5, the Stein operator for the standard normal distri-*

[1] For example, the $\mathcal{X}^2$ test is restricted to univariate distributions.

[2] This would only be possible if we had explicit access to the distribution Q, which is rarely the case. Instead, we require a notion of discrepancy between a sample $\{x_i\} \sim q(x)$ and the target distribution.

[3] There are some technicalities that we do not discuss here. See Liu *et al.* (2016) for a goodness-of-fit testing procedure based on kernelised Stein discrepancy, which may potentially be generalised to other discrepancy measures.

[4] In the case of a scalar-valued function $f : \mathcal{X} \to \mathbb{R}$, the Langevin-Stein operator is given by $\mathcal{T}_p f(x) = s_p(x) f(x) + \nabla_x f(x)$. Here, ∇f and $\mathcal{T}_p f$ are $d \times 1$ vector-valued functions.

*bution is a **Langevin-Stein** operator. This can easily be seen by writing:*

$$\mathcal{T}_\phi f(z) = \frac{\nabla_z \phi(z)}{\phi(z)} f(z) + \nabla_z f(z) = \frac{-z\phi(z)}{\phi(z)} f(z) + \nabla_z f(z) = \nabla_z f(z) - z f(z)$$

Here we have that $z \in \mathbb{R}$ and hence the gradient $\nabla_z f(z)$ is replaced by the derivative $f'(z)$, which then yields the Stein operator for the standard normal distribution as given in Example 2.5.

Lemma 2.10. (Score function)
Let P be a distribution with continuous density function p. The score function $s_p(x) = \nabla_x \log p(x)$ is independent of the normalisation constant of p. I.e. let $p(x) = \frac{1}{Z}\tilde{p}(x)$ where Z denotes the normalisation constant, then we have that $s_p = s_{\tilde{p}}$.

Proof. See Appendix A. □

The fact that the score function $s_p(x)$ is independent of the normalisation constant of p is an extremely important and useful result since it is often the case in machine learning problems that we only have access to an unnormalised target density. As we will show, this result enables the computation of kernelised Stein discrepancy and the implementation of Stein variational gradient descent for unnormalised target densities.

We now formally define the Stein class corresponding to the Langevin-Stein operator in Definition 2.11, which is taken from Liu *et al.* (2016).

Definition 2.11. (Langevin-Stein Class)
*A scalar-valued function $f : \mathcal{X} \to \mathbb{R}$ is said to be in the **Langevin-Stein class** of a distribution P with density p if f is continuously differentiable and satisfies the following:*

$$\int_{x \in \mathcal{X}} \nabla_x \left[f(x) p(x) \right] dx = 0 \ . \tag{2.9}$$

A vector-valued function $f : \mathcal{X} \to \mathbb{R}^{d'}$ of the form $f(x) = \begin{bmatrix} f_1(x) & f_2(x) & \dots & f_{d'}(x) \end{bmatrix}^T$ is said to be in the Langevin-Stein class of P if each component function f_i is in the Langevin-Stein class of P.

Corollary 2.12. *Definition 2.11 of the Langevin-Stein class coincides with the general definition of a Stein class in Definition 2.4 when using the Langevin-Stein operator. I.e. $\mathbb{E}_{X \sim P}\left[\mathcal{T}_p f(X)\right] =$*

$0 \iff \int_{x\in\mathcal{X}} \nabla_x [f(x)p(x)]\, dx = 0$ *when* $\mathcal{T}_p$ *is the Langevin-Stein operator.*

Proof. See Appendix A. □

We are now in a position to discuss the so-called *Stein identity*, which forms the basis for the Stein discrepancy measure (see Stein *et al.* (2004) and Chen (2021), for example, for more information).

Lemma 2.13. (Stein's identity)

Let P *be a distribution with a continuous, differentiable density* p*, and let* $\mathcal{T}_p$ *be its associated Langevin-Stein operator. Then, for any vector-valued function* $f \in \mathcal{F}(\mathcal{T}_p)$ *in the Langevin-Stein class of* P*, we have that* [5]*:*

$$\mathbb{E}_{X\sim P}\left[\mathcal{T}_p f(X)\right] := \mathbb{E}_{X\sim P}\left[s_p(X)f(X)^T + \nabla_X f(X)\right] = 0 \ . \tag{2.10}$$

Proof. See Appendix A. □

Remark: Stein's identity (Lemma 2.13) is a generalisation of Stein's lemma (Lemma 2.1) to distributions beyond the standard normal distribution.

Corollary 2.14. *(Ley & Swan, 2013)*

Let P*,* p*, and* $\mathcal{T}_p$ *be defined as in Lemma 2.13. Furthermore, let* $Q \neq P$ *be a distribution supported on* $\mathcal{X}$ *with continuous, differentiable density* $q(x)$*, and suppose that Stein's identity holds for* Q *with Langevin-Stein operator* $\mathcal{T}_q$*, i.e.* $\mathbb{E}_{X\sim Q}\left[\mathcal{T}_q f(X)\right] = 0$ *for all functions* f *in the Langevin-Stein class of* Q*. Furthermore, let* $f : \mathcal{X} \to \mathbb{R}^{d'}$ *be a vector-valued function in the Langevin-Stein class of* P*. If we consider Stein's identity in Equation (2.10) when taking expectations with respect to* Q*, we can write:* [6]

$$\mathbb{E}_{X\sim Q}\left[\mathcal{T}_p f(X)\right] = \mathbb{E}_{X\sim Q}\left[\left(s_p(X) - s_q(X)\right) f(X)^T\right] \ . \tag{2.11}$$

Proof. See Appendix A. □

Theorem 2.15. (Toward Stein discrepancy)

Consider the same setting as in Corollary 2.14. If $P \neq Q$ *then there must exist a function* f *such*

[5] In the case of scalar-valued functions $f \in \mathcal{F}(\mathcal{T}_p)$, a similar results to Equation (2.10) holds by omitting the transpose on f.

[6] In the case of a scalar-valued function $f : \mathcal{X} \to \mathbb{R}$, a similar result to Equation (2.11) holds by simply omitting the transpose.

that the quantity in Equation (2.11) is non-zero. I.e.,

$$P \neq Q \implies \exists f \;\; s.t. \quad \mathbb{E}_{X \sim Q}\left[\mathcal{T}_p f(X)\right] \neq 0 \; . \tag{2.12}$$

Proof. See Appendix A. □

Theorem 2.15 suggests that we may be able to compare two distributions by considering the amount by which $\mathbb{E}_{X \sim Q}\left[\mathcal{T}_p f(X)\right]$ deviates from zero. Since this quantity depends on the choice of function f, it is natural to seek the function f that yields the "maximum violation of Stein's identity" (Liu & Feng, 2017). This gives rise to the notion of a Stein discrepancy.

Definition 2.16. (Stein discrepancy)
Let P and Q be distributions with continuous, differentiable density functions p and q respectively. Let $\mathcal{T}_p$ be the Langevin-Stein operator characterising the distribution P, and let $\mathcal{F}_q$ be a class of continuously differentiable vector-valued functions that satisfy Stein's identity in Equation (2.10). The Stein discrepancy from q to p with respect to $\mathcal{F}_q$ is defined by Liu & Wang (2016) as: [7]

$$\mathbb{D}_{\text{Stein}}(q, p; \mathcal{F}_q) := \sup_{f \in \mathcal{F}_q} \mathbb{E}_{X \sim Q}\left[\operatorname{trace}\left(\mathcal{T}_p f(X)\right)\right]^2 = \sup_{f \in \mathcal{F}_q} \mathbb{E}_{X \sim Q}\left[\left(s_p(X) - s_q(X)\right)^T f(X)\right]^2 \; . \tag{2.13}$$

Remark: The class of functions $\mathcal{F}_q$ in Definition 2.16 should be chosen to be rich enough such that the resulting Stein discrepancy is a valid discrepancy measure, i.e. so that $\mathbb{D}_{\text{Stein}}(q, p; \mathcal{F}_q) > 0$ whenever $p \neq q$ (Liu *et al.*, 2016). However, the function class should also be chosen such that $\mathbb{D}_{\text{Stein}}(q, p; \mathcal{F}_q)$ is computationally tractable [8].

Given a sufficiently rich class of functions $\mathcal{F}_q$, the GOF test in Equation (2.6) reduces to testing $H_0 : \mathbb{D}_{\text{Stein}}(q, p; \mathcal{F}_q) = 0$ [9]. However, there are two caveats to this approach that limit its use in practice: firstly, we have to explicitly specify the function class $\mathcal{F}_q$ over which the optimisation is performed, and secondly, computing the Stein discrepancy may still be computationally intractable.

[7] This form of Stein discrepancy is the square of the typical definition given by e.g. Gorham *et al.* (2019). We use the squared definition since it is more closely related to the definition of kernelised Stein discrepancy in Equation (2.14).

[8] These desiderata are closely related to those in variational inference where we seek a class of distributions $\mathcal{Q}$ that is rich enough to enable accurate approximation of a target distribution P, but not too rich such that the variational optimisation is intractable.

[9] This would again only be possible if we had explicit access to the true distribution $q(x)$, which is rarely the case. Here we require a notion of discrepancy between a sample $\{x_i\} \sim q(x)$ and the target distribution, i.e. $\mathbb{D}(\{x_i\}||p)$.

In the following section, we discuss the particular choice of taking $\mathcal{F}_q$ to be the unit ball in a reproducing kernel Hilbert space. This gives rise to the *kernelised Stein discrepancy* proposed independently by Liu *et al.* (2016) and Chwialkowski *et al.* (2016).

2.5 KERNELISED STEIN DISCREPANCY

This section discusses the kernelised Stein discrepancy (KSD) (Liu *et al.*, 2016; Chwialkowski *et al.*, 2016), a tractable form of Stein discrepancy that takes the function class $\mathcal{F}_q$ in Definition 2.16 to be the unit ball in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. As will soon become apparent, this choice overcomes both limitations discussed at the end of the previous section. Firstly, this choice ensures a sufficiently rich class of functions $\mathcal{F}_q$ since an RKHS is (often) infinite-dimensional (Ghojogh *et al.*, 2021); secondly, the resulting Stein discrepancy is computationally tractable, and can be computed in closed form (Liu *et al.*, 2016) via a special kind of "kernel trick", which we will refer to as a "Steinalised kernel trick" in connection to the "Steinalized" kernel (Liu, 2017) in Equation (2.16).

Notations We assume the same notations as Liu & Wang (2016), which we describe here. Denote by $\mathcal{H}_0$ an RKHS of real-valued functions $f : \mathcal{X} \to \mathbb{R}$ associated with a positive definite reproducing kernel $k(x, x')$, and let $\langle \cdot, \cdot \rangle_{\mathcal{H}_0}$ denote the corresponding inner product. We extend the RKHS $\mathcal{H}_0$ to an RKHS of vector-valued functions $f : \mathcal{X} \to \mathbb{R}^{d'}$, which is given by the Cartesian product of d' copies of $\mathcal{H}_0$, i.e. $\mathcal{H} = \mathcal{H}_0 \times \cdots \times \mathcal{H}_0$. A function $f(x) = \begin{bmatrix} f_1(x) & f_2(x) & \dots & f_{d'}(x) \end{bmatrix}^T$ is in $\mathcal{H}$ if each f_i is in $\mathcal{H}_0$. The inner product on $\mathcal{H}$ is defined as: $\langle f, g \rangle_{\mathcal{H}} = \sum_{l=1}^{d'} \langle f_l, g_l \rangle_{\mathcal{H}_0}$. Let $\|\cdot\|_{\mathcal{H}}$ denote the norm induced by this inner product. Finally, a (closed) ball of radius δ in $\mathcal{H}$ is given by $\mathcal{B}_{\mathcal{H}} = \{f \in \mathcal{H} : \|f\|_{\mathcal{H}}^2 \leq \delta\}$.

With these notations at hand, we define the KSD as in Liu *et al.* (2016).

Definition 2.17. (Kernelised Stein discrepancy) *(Liu et al., 2016)*
Let P and Q be two distributions supported on a measurable space $(\Omega, \mathcal{X}), \mathcal{X} \subseteq \mathbb{R}^d$ with continuously differentiable densities $p(x)$ and $q(x)$ respectively. Furthermore, let $k(x, x')$ be a positive definite

kernel. The ***kernelised Stein discrepancy*** *(KSD) between distributions P and Q is given by:* [10]

$$\mathbb{S}(q||p) = \mathbb{E}_{X,X'\sim Q}\left[\left(s_p(X) - s_q(X)\right)^T k(X,X')\left(s_p(X') - s_q(X')\right)\right] \tag{2.14}$$

Liu *et al.* (2016:Proposition 3.3) show that, provided the kernel $k(x,x')$ is integrally strictly positive definite (see Definition 3.1 of Liu *et al.* (2016)), and under some mild assumptions about the densities p and q, the KSD is a valid discrepancy measure [11]. This means that $\mathbb{S}(q||p) = 0 \iff p = q$. Liu *et al.* (2016) mention that the assumptions may be violated when $q(x)$ has a heavy tail [12], in which case KSD may fail to be a valid discrepancy measure.

The form of KSD given in Equation (2.14) is problematic since it requires the calculation of the score function of the unknown distribution $q(x)$, which cannot be computed exactly since we only have access to the distribution via a sample $\{x_i\} \overset{i.i.d}{\sim} q(x)$ [13]. Liu *et al.* (2016) overcome this problem by using Theorem 1 of Oates *et al.* (2017) to obtain a form of KSD that only requires the score function of the target distribution P. Their result relies on the assumption that the kernel $k(x,x')$ is in the Stein class of Q in the sense given below.

Definition 2.18. *(Liu et al., 2016)*
A kernel $k(x,x')$ is said to be in the Stein class of Q if $k(x,x')$ has continuous second-order partial derivatives, and both $k(x,\cdot)$ and $k(\cdot,x)$ are in the Stein class of Q for all fixed $x \in \mathcal{X}$.

Remark: Liu *et al.* (2016:Proposition 3.5) show that if the kernel $k(x,x')$ with corresponding RKHS $\mathcal{H}$ is in the Stein class of Q, then so is any $f \in \mathcal{H}$.

Liu *et al.* (2016:Theorem 3.6) show that, if $k(x,x')$ is in the Stein class of Q, then the KSD is also given by: [14]

$$\mathbb{S}(q||p) = \mathbb{E}_{X,X'\sim Q}\left[\kappa_p(X,X')\right] \quad , \tag{2.15}$$

[10] See Appendix A for a derivation of this form of KSD starting from the general definition of Stein discrepancy in Definition 2.16.

[11] Chwialkowski *et al.* (2016:Theorem 2.2) and Barp *et al.* (2022:Proposition 1) also prove that the KSD is a valid discrepancy measure based on different sets of assumptions.

[12] See South *et al.* (2022) for a tail condition on $q(x)$ for Stein's identity to hold.

[13] However, it may be possible to estimate the score function using score matching techniques. See Hyvärinen (2005) for a discussion on score function estimation.

[14] An equivalent result is given in Theorem 2.1 of Chwialkowski *et al.* (2016).

where $\kappa_p(x, x')$ is called a "Stein kernel" (Kanagawa *et al.*, 2023) and is given by:

$$\begin{aligned}\kappa_p(x, x') = s_p(x)^T k(x, x') s_p(x') &+ s_p(x)^T \nabla_{x'} k(x, x') \\ &+ \nabla_x k(x, x')^T s_p(x') + \mathrm{trace}\left(\nabla_x \nabla_{x'} k(x, x')\right) \quad .\end{aligned} \tag{2.16}$$

Remark: The kernel above, referred to as a "Steinalized" kernel by Liu (2017), can be obtained by applying the Langevin-Stein operator to the kernel $k(x, x')$ twice, once for each argument (Liu *et al.*, 2016), i.e. $\kappa_p(x, x') = \mathcal{T}_p^x \left(\mathcal{T}_p^{x'} k(x, x')\right)$, where $\mathcal{T}_p^x$ denotes the Langevin-Stein operator with respect to x.

In practice, the KSD in Equation (2.15) can be estimated via a *U-statistic* (Hoeffding, 1948):

$$\hat{\mathbb{S}}(q||p) = \hat{\mathbb{E}}_{X,X' \sim Q}\left[\kappa_p(X, X')\right] = \frac{1}{n(n-1)} \sum_{i=1,}^{n} \sum_{j \neq i} \kappa_p(x_i, x_j) \tag{2.17}$$

where $x_i \overset{i.i.d}{\sim} q(x), i = 1, 2, \ldots, n$.

This is a powerful result that enables non-parametric GOF tests of the hypothesis test in Equation (2.6) that does not require explicit access to the true distribution $q(x)$, nor does it require samples from the target distribution $p(x)$. Hence, we now have a notion of a discrepancy, $\mathbb{S}(\{x_i\}||p)$, between a sample $\{x_i\} \overset{i.i.d}{\sim} q(x)$ and the target distribution $p(x)$. Consequently, this gives rise to a framework for conducting GOF tests using only a sample $\{x_i\} \overset{i.i.d}{\sim} q(x)$, and the score function, s_p, of the target distribution $p(x)$.

2.6 STEIN VARIATIONAL GRADIENT DESCENT

The previous section discussed how the kernelised Stein discrepancy can be used to quantify the similarity/dissimilarity between two distributions P and Q. This section discusses the Stein Variational Gradient Descent (SVGD) algorithm as proposed by Liu *&* Wang (2016), which is used to incrementally transform the distribution Q into P by performing functional gradient descent on the Kullback-Leibler (KL) divergence (Liu *&* Wang, 2016; Han *&* Liu, 2018). Before presenting the SVGD algorithm, we first provide some background on variational inference, specifically variational inference with smooth transforms. For a detailed discussion on variational inference, see Blei *et al.*

(2017).

2.6.1 Background and Notation

Notations For the remainder of this section, we assume that we are working with a continuous random variable X defined on a measurable space $(\Omega, \mathcal{X}), \mathcal{X} \subseteq \mathbb{R}^d$. Furthermore, we assume that we are interested in approximating an (intractable) target distribution P supported on $\mathcal{X}$ with a continuous, differentiable density $p(x)$. We further assume the same notations as in Section 2.5.

Variational Inference Variational inference (VI) refers to approximating a target distribution by finding the *closest* distribution within a predetermined family of distributions. That is, suppose we are tasked with estimating a difficult (or possibly intractable) to compute distribution $p(x)$ - a common case is the posterior distribution in Bayesian inference that (typically) cannot be evaluated in closed form. VI addresses this problem by positing a family of tractable distributions $\mathcal{Q}$ and proceeds to find the distribution $q^* \in \mathcal{Q}$ that provides the best approximation of the target distribution p. In this way, VI turns the problem of estimating an arbitrary target distribution into an optimisation problem, where we seek the distribution $q \in \mathcal{Q}$ that minimises (typically) the (reverse) KL divergence to the target distribution. Therefore, VI is typically concerned with the following optimisation problem:

$$\begin{aligned} q^* = \arg\min_{q \in \mathcal{Q}} \mathbb{D}_{\mathrm{KL}}(q||p) &= \arg\min_{q \in \mathcal{Q}} \mathbb{E}_{X \sim Q}\left[\log q(X) - \log \tilde{p}(X) + \log Z_p\right] \\ &\equiv \arg\min_{q \in \mathcal{Q}} \mathbb{E}_{X \sim Q}\left[\log q(X) - \log \tilde{p}(X)\right] \end{aligned}$$

where $\tilde{p}$ denotes the unnormalised target density and Z_p its normalisation constant.

It is common to take the variational family $\mathcal{Q}$ to be a parameterised family of distributions, indexed by a parameter (vector) $\theta \in \Theta$, where the parameters θ are called the free variational parameters. In this case, the variational optimisation problem can be rephrased as finding the optimal parameters $\theta^* \in \Theta$ to provide the best approximation of the target distribution, i.e.

$$\theta^* = \arg\min_{\theta \in \Theta} \mathbb{D}_{\mathrm{KL}}(q(x;\theta)||p(x)) \ .$$

Variational Inference with Smooth Transforms In VI with smooth transforms, we take the family of distributions $\mathcal{Q}$ to consist of distributions obtained via smooth transforms from a tractable reference distribution. Given a parameterised family of smooth transforms (continuous, differentiable bijections with differentiable inverses) $T_\psi : \mathcal{X} \to \mathcal{X}$, and a tractable reference distribution $q_0(x)$, we take $\mathcal{Q}$ to be the set of distributions of the random variable $Z = T_\psi(X)$, with $X \sim q_0(x)$. The distribution of $Z = T_\psi(X)$ can be obtained via the change-of-variables formula (e.g., Papamakarios *et al.*, 2021):

$$q_{[\psi]}(z) = q_0(T_\psi^{-1}(z)) \cdot |\det J_{T_\psi^{-1}}(z)| \tag{2.18}$$

where $J_{T_\psi^{-1}}$ denotes the Jacobian of the inverse transform T_ψ^{-1}.

2.6.2 Stein Variational Gradient Descent

In this section, we review the original formulation of SVGD, which we refer to as vanilla SVGD. SVGD (Liu & Wang, 2016) is a non-parametric, particle-based variational inference algorithm that enables approximate inference for (or sampling from) intractable target distributions. It works by iteratively transporting a set of particles to approximate the target distribution (Liu & Wang, 2016; Liu, 2017). In each iteration, the update directions of the particles are chosen to yield the maximum reduction in (reverse) KL divergence between the distribution represented by the particles and the target distribution. As a result, SVGD can be viewed as “a type of functional gradient descent on the KL divergence” (Han & Liu, 2018).

SVGD starts with an initial set of particles, $\{x_i\}_{i=1}^n$, drawn from an arbitrary, simple initial distribution $q_0(x)$ (e.g. the standard normal distribution), and iteratively updates the positions of the particles via a perturbed identity map (Liu & Wang, 2016) given by:

$$T(x) := x + \epsilon\phi(x) \tag{2.19}$$

where $\epsilon > 0$ is a step size and $\phi(x)$ is a velocity field that determines the update direction. In each iteration, the velocity field ϕ is chosen from a suitable class of continuously differentiable functions $\mathcal{F}$ to maximise the reduction in (reverse) KL divergence[15] (Liu, 2016; Liu & Wang, 2016). Specifically,

$$\phi(x) = \arg\max_{\phi\in\mathcal{F}} \left\{ -\frac{d}{d\epsilon}\mathbb{D}_{\mathrm{KL}}(q_{[\epsilon\phi]}||p)\big|_{\epsilon=0} \right\} \tag{2.20}$$

where $q_{[\epsilon\phi]}$ denotes the density function represented by the updated particles, $x' = x + \epsilon\phi(x)$, with $x \sim q(x)$ [16].

Liu & Wang (2018) decompose the KL divergence as:

$$\mathbb{D}_{\mathrm{KL}}(q_{[\epsilon\phi]}||p) = \mathbb{D}_{\mathrm{KL}}(q||p) - \epsilon\mathbb{E}_{X\sim Q}\left[\mathrm{trace}(\mathcal{T}_p\phi(X))\right] + O(\epsilon^2) \ .$$

Given this decomposition of the KL divergence, it is straightforward to arrive at the relationship between the derivative of the KL divergence and the Langevin-Stein operator as given by Theorem 3.1 of Liu & Wang (2016):

$$-\frac{d}{d\epsilon}\mathbb{D}_{\mathrm{KL}}(q_{[\epsilon\phi]}||p)\big|_{\epsilon=0} = \mathbb{E}_{X\sim Q}\left[\mathrm{trace}\left(\mathcal{T}_p\phi(X)\right)\right] \ . \tag{2.21}$$

In this way, the functional optimisation problem in Equation (2.20) can be rewritten as:

$$\phi(x) = \arg\max_{\phi\in\mathcal{F}} \left\{\mathbb{E}_{X\sim Q}\left[\mathrm{trace}(\mathcal{T}_p\phi(X))\right]\right\}$$

where the maximum reduction in the (reverse) KL divergence can now be related to the Stein discrepancy of Equation (2.13) via:

$$\max_{\phi\in\mathcal{F}} \left\{ -\frac{d}{d\epsilon}\mathbb{D}_{\mathrm{KL}}(q_{[\epsilon\phi]}||p)\big|_{\epsilon=0} \right\} = \sqrt{\mathbb{D}_{\mathrm{Stein}}(q,p;\mathcal{F})} \ .$$

Unfortunately, this approach suffers from the same limitations as the Stein discrepancy discussed in Section 2.4. To overcome these limitations, Liu & Wang (2016) once again take $\mathcal{F}$ to be the closed ball in an RKHS $\mathcal{H}$ corresponding to a positive definite kernel $k(x, x')$, given by $\mathcal{B} = \{\phi \in \mathcal{H} : \|\phi\|^2_{\mathcal{H}} \leq \mathbb{S}(q||p)\}$, similar to how KSD overcomes the limitations of Stein discrepancy, but using a different radius for the closed ball $\mathcal{B}$. In this case, Liu & Wang (2016:Lemma 3.2) show that the

[15] Technically, the update directions are chosen to maximise the rate of decay in KL divergence.

[16] If $|\epsilon|$ is sufficiently small such that the perturbed identity map is invertible (Liu & Wang, 2016), $q_{[\epsilon\phi]}$ can be obtained via the change-of-variables formula.

optimal update directions can be computed in closed form by:

$$\phi^*(\cdot) = \mathbb{E}_{X\sim Q}\left[\mathcal{T}_p k(X,\cdot)\right] = \mathbb{E}_{X\sim Q}\left[k(X,\cdot)\nabla_X \log p(X) + \nabla_X k(X,\cdot)\right] \tag{2.22}$$

for which the (squared) decrease in (reverse) KL divergence is exactly equal to the KSD,

$$-\frac{d}{d\epsilon}\mathbb{D}_{\mathrm{KL}}(q_{[\epsilon\phi^*]}||p)\big|_{\epsilon=0} = \sqrt{\mathbb{S}(q||p)}\ . \tag{2.23}$$

In practice, empirical averaging is used to estimate the expectation under the current distribution Q, with density function q, represented by the current particles (Liu, 2016), which then yields an estimate of the optimal update direction given by:

$$\begin{aligned}\hat{\phi}^*(\cdot) &= \hat{\mathbb{E}}_{X\sim Q}\left[k(X,\cdot)s_p(X) + \nabla_X k(X,\cdot)\right] \\ &= \frac{1}{n}\sum_{j=1}^{n}\left[k(x_j,\cdot)\nabla_{x_j}\log p(x_j) + \nabla_{x_j}k(x_j,\cdot)\right]\end{aligned} \tag{2.24}$$

where $x_i \overset{i.i.d}{\sim} q(x), i = 1, 2, \dots, n$.

Given the optimal update direction above, SVGD iteratively updates the positions of the particles $\{x_i\}_{i=1}^n$ via the update equation (obtained by setting $\phi(x)$ in Equation (2.19) according to Equation (2.24)) given by:

$$x_i \leftarrow x_i + \frac{\epsilon}{n}\sum_{j=1}^{n}\underbrace{k(x_j,x_i)\nabla_{x_j}\log p(x_j)}_{\text{driving force}} + \underbrace{\nabla_{x_j}k(x_j,x_i)}_{\text{repulsive force}}, \quad i = 1, 2, \dots, n. \tag{2.25}$$

Remark: The update rule in Equation (2.25) contains two opposing terms: the first term is a kernel-weighted gradient of the log density of the target distribution, which serves as a **driving force** (e.g., Liu *&* Wang, 2016; Zhou *&* Qiu, 2023) that pushes the particles towards high-density regions of the target distribution, with information sharing across the particles via the weighting by kernel similarities; the second term is the gradient of the kernel function, which serves as a **repulsive force** (e.g., Liu *&* Wang, 2016; Ba *et al.*, 2022) that pushes the particles away from each other, thereby encouraging diversity in the particle positions to prevent the particles from collapsing into a single mode of the target density. Furthermore, the relative magnitude of the deterministic repulsive force helps to ensure that the particle diversity accurately reflects the uncertainty (i.e., the variance) in the target distribution.

The vanilla SVGD algorithm is summarised in Algorithm 1.

Illustrative example of SVGD To illustrate the usage of vanilla SVGD, we consider a simple sampling experiment where the goal is to sample from a bivariate, two-component Gaussian mixture model (GMM) target distribution given by:

$$p(x) = 0.5\mathcal{N}\left(x; \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.52 & 0.92 \\ 0.92 & 3.05 \end{bmatrix}\right) + 0.5\mathcal{N}\left(x; \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.52 & 0.92 \\ 0.92 & 3.05 \end{bmatrix}^{-1}\right) .$$

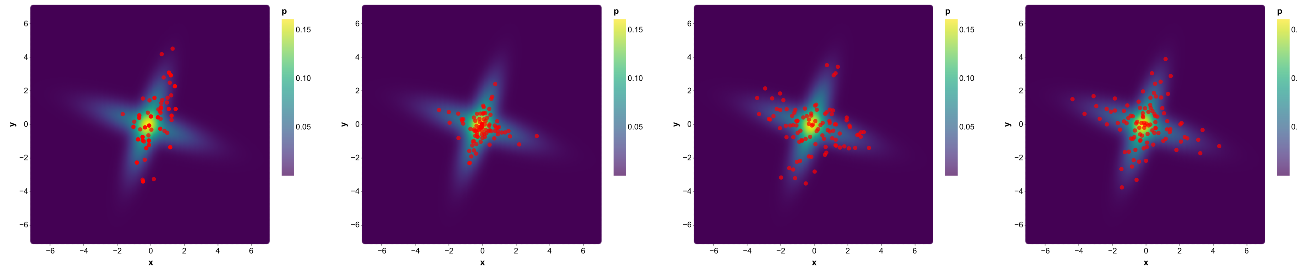
We sample from this target distribution using several sampling algorithms: the Random-Walk Metropolis-Hastings (RW-MH) sampler, the Hamiltonian/Hybrid Monte Carlo (Duane *et al.*, 1987) (HMC) sampler, the No U-Turn Sampler (Hoffman *&* Gelman, 2011) (NUTS), and finally SVGD [17]. We visualise the sampled points in Figure 2.1 by plotting them on the density contours of the GMM. Furthermore, we estimate the KSD between each of the samples and the target distribution and summarise the results in Table 2.1. The results indicate that SVGD yields the most accurate sample from the GMM since it (i) provides the best coverage of the target probability space as evident in Figure 2.1, and (ii) has the smallest estimated KSD as evident in Table 2.1 [18].

Algorithm	Estimated KSD
MH	0.1401
HMC	0.3233
NUTS	0.0453
SVGD	-0.0713

Table 2.1: Estimated kernelised Stein discrepancy for each of the samples.

[17] We implement the RW-MH, HMC and SVGD samplers from scratch, while we use the TensorFlow Probability package (Abadi *et al.*, 2015) to implement NUTS.

[18] Note that, although KSD is non-negative in theory, errors introduced through numerical approximation sometimes results in negative estimates.



(a) RW-MH (b) HMC (c) NUTS (d) SVGD

Figure 2.1: Sampling from a bivariate GMM. We sample from the target distribution using four samplers: random-walk MH (first), HMC (second), NUTS (third), and SVGD (fourth). We plot the samples on the density contours of the GMM. For SVGD, we use an RBF kernel with median heuristic bandwidth - see Equation 4.10.

2.7 CONVERGENCE PROPERTIES OF SVGD

Since the SVGD algorithm was first proposed, there have been several works aimed at a convergence analysis of SVGD in the mean-field limit [19] (e.g., Liu, 2017; Lu *et al.*, 2019; Gorham *et al.*, 2020; Salim *et al.*, 2022). In this section, we provide a high-level overview of the ideas used by Liu (2017) to prove that, in the mean-field limit, particles evolving according to SVGD dynamics converge to the target distribution, provided that the step sizes ϵ_t are decreased sufficiently fast and that the initial distribution q_0 has finite (reverse) KL divergence with the target distribution (Liu, 2017).

Notations Let P be the target probability measure supported on measurable space $\mathcal{X} \subseteq \mathbb{R}^d$ with a continuous, differentiable density function $p(x)$. Furthermore, let $\hat{Q}_t^n(dx) = \frac{1}{n}\sum_{i=1}^n \delta(x - x_i^{(t)})dx$ denote the empirical measure of the n particles $\{x_i^{(t)}\}_{i=1}^n$ at the t^{th} iteration of SVGD. Liu (2017) defines the map $\Phi_p : Q \mapsto T_\# Q$, which maps the measure Q to the pushforward measure $T_\# Q$ through the map in Equation (2.19) with the optimal perturbation direction ϕ^* given in Equation (2.22). This map characterises the evolution of the empirical measure of the particles in the sense that $\hat{Q}_{t+1}^n = \Phi_p(\hat{Q}_t^n)$ $\forall t \in \mathbb{N}$ (Liu, 2017).

Liu (2017) first considers the many-particle asymptotic behaviour of SVGD under the assumption that the limit initial measure $\hat{Q}_0^n$ weakly converges to some measure Q_0^∞ as $n \to \infty$ [20]. Assuming

[19] The mean-field limit refers to the infinite-particle limit under a fixed dimensionality.

[20] As discussed by Liu (2017), this can easily be achieved by taking the initial particles $\{x_i^{(0)}\}$ to be an i.i.d sample from distribution q_0 corresponding to the measure Q_0^∞.

that $\hat{Q}_0^n \rightharpoonup Q_0^\infty$ as $n \to \infty$, and assuming an appropriate Lipschitz condition on the map Φ_p, Liu (2017) shows that $\mathrm{BL}(\hat{Q}_t^n, Q_t^\infty) \to 0$ as $n \to \infty \ \forall t \in \mathbb{N}$, where $\mathrm{BL}(\cdot, \cdot)$ denotes the bounded Lipschitz (BL) metric. Since the BL metric is known to metricise weak convergence (e.g., Budhiraja *et al.*, 2012; Lunde, 2019), this result implies that $\hat{Q}_t^n \rightharpoonup Q_t^\infty \ \forall t \in \mathbb{N}$. Now it only remains to show that $Q_t^\infty \rightharpoonup P$ as $t \to \infty$. Intuitively, since the SVGD algorithm is guaranteed to monotonically decrease the KL divergence between the particle distribution and the target distribution in each iteration (Liu, 2017:Theorem 3.3(2)), and since the KL divergence metricises weak convergence (e.g., Walker, 2004; Pinski *et al.*, 2015), the limit empirical measure Q_t^∞ weakly converges to the target measure P as $t \to \infty$. Therefore, Liu (2017) concludes that $\hat{Q}_t^n \rightharpoonup P$ as $n \to \infty, t \to \infty$, thereby establishing convergence of particles evolving according to SVGD dynamics to the target distribution.

Although convergence is guaranteed in the infinite-time limit when using infinitely many particles, convergence is not guaranteed in practice due to limited time and memory. Furthermore, there is limited work on proving convergence (and establishing explicit convergence rates) of SVGD in the finite-particle and finite-time regime, with the exception of Shi *&* Mackey (2022) who derive explicit convergence rates under stringent assumptions on the target distribution.

2.8 CONCLUSION

This chapter developed the vanilla SVGD algorithm, starting from Stein's method for bounding reference IPMs in Section 2.3, moving to Stein discrepancy in Section 2.4, and its kernelised variant, KSD, in Section 2.5. Section 2.6 presented the vanilla SVGD algorithm along with a simple sampling application thereof. The sampling experiment demonstrated the effectiveness of SVGD, where it was able to outperform three well-known MCMC algorithms in terms of both sampling accuracy and speed. Finally, Section 2.7 provided a brief overview of the convergence properties of SVGD as given by Liu (2017), who showed that the empirical measure of particles evolving according to SVGD dynamics weakly converges to the target measure.

The following chapter discusses the major advantages and limitations of vanilla SVGD, aiming to shed further light on the practical applicability of SVGD by considering scenarios in which SVGD may fail to accurately approximate the target distribution.

CHAPTER 3

MAJOR ADVANTAGES AND LIMITATIONS OF VANILLA SVGD

3.1 INTRODUCTION

This chapter discusses the major advantages and limitations of vanilla SVGD. Furthermore, a brief comparison between SVGD and alternative inference/sampling methods is given, together with a discussion of several improvements to vanilla SVGD.

3.2 ADVANTAGES AND COMPARISON TO OTHER APPROACHES

Advantages relative to MCMC and VI

Stein Variational Gradient Descent (SVGD) integrates key benefits from both variational inference (VI) and Markov Chain Monte Carlo (MCMC) methods (e.g., Yan *&* Zhou, 2021; Ai *et al.*, 2022). Whilst MCMC methods are guaranteed to be asymptotically correct (e.g., Salimans *et al.*, 2015), the auto-correlation between successive sample points often results in slow convergence in practice (Robert *et al.*, 2018; Zhang *et al.*, 2019), which necessitates simulating long Markov chains to achieve high accuracy. Conversely, VI methods are generally much faster than MCMC methods (Gunapati *et al.*, 2022; Ganguly *&* Earp, 2021), but are usually not asymptotically correct (Blei *et al.*, 2017). This means that, in most cases, the variational distribution will not even asymptotically match the target distribution.

SVGD lies somewhere in the middle between MCMC and VI methods (Detommaso *et al.*, 2018; Pinder *et al.*, 2020), and can be viewed as either a non-parametric VI algorithm or a deterministic, particle-based sampling algorithm (Ai *et al.*, 2022). On the one hand, like other VI methods, SVGD is generally faster than MCMC methods due to the deterministic nature of its updates and efficient use of gradient information (e.g., Yoon *et al.*, 2018)[1], and since SVGD does not require a *burn-in* phase as do many MCMC methods. On the other hand, since SVGD is non-parametric and does not involve any assumptions on the form of the target distribution, it potentially allows a more

[1] Note that, some MCMC methods such as Hamiltonian Monte Carlo, also use gradient information, but rely on randomness in updating sample points.

accurate approximation of the target distribution compared to traditional VI methods, and has the added benefit of being asymptotically correct in the sense of weak convergence to the target distribution (see Liu (2017) and Lu *et al.* (2019), for example, for proofs of convergence in the mean-field limit).

Another distinguishing property of SVGD in the context of sampling is that SVGD evolves a set of particles simultaneously, whereas MCMC methods generate samples sequentially (Ye *et al.*, 2020). Furthermore, SVGD does not involve rejecting proposed sample points and hence the effective number of particles is equal to the number of initial particles (Han *&* Liu, 2018).

Other Advantages

A defining feature of SVGD is that it provides a spectrum of inference algorithms depending on the number of particles used. When using only a single particle ($n = 1$), and a kernel that satisfies $\nabla_x k(x, x') = 0$ whenever $x = x'$, then SVGD reduces to gradient ascent for maximum *a posteriori* (MAP) estimation [2] (Liu *&* Wang, 2016). Conversely, in the limit of infinitely many particles ($n \to \infty$), SVGD becomes a full Bayesian inference algorithm (Liu *&* Wang, 2016). Hence, SVGD is considered to be more particle-efficient than MCMC methods since it can achieve good results with relatively few particles (e.g., Liu *&* Zhu, 2017; Das *&* Nagaraj, 2023). In addition to being particle-efficient, SVGD is also considered to be iteration-efficient (Liu *&* Zhu, 2017) since it is guaranteed to make progress in every iteration in the sense of decreasing the KL divergence (Liu, 2017:Theorem 3.3(2)). Lastly, (vanilla) SVGD (and variants thereof) are very versatile, having been applied to several complex problems such as training a Generative Adversarial Network (GAN) (Wang *et al.*, 2022), training a Variational Autoencoder (VAE) (Pu *et al.*, 2017), and training Bayesian Neural Networks (BNNs) (e.g., Liu *&* Wang, 2016). Moreover, as we discuss in Chapter 4, a technique based on SVGD has been applied in reinforcement learning for learning a diverse set of policies (Liu *et al.*, 2017).

[2] This is true assuming the target distribution $p(x)$ represents a posterior distribution, otherwise SVGD reduces to gradient ascent for maximum likelihood estimation.

3.3 LIMITATIONS AND IMPROVEMENTS

Whilst SVGD has the potential to accurately approximate complex target distributions in certain cases, various challenges exist that inhibit the widespread use of SVGD in practice. This section discusses the major limitations of SVGD, shedding light on the applicability of SVGD in practice.

Variance collapse

The major limitation of SVGD is the so-called *variance collapse* phenomenon (Ba *et al.*, 2022), also referred to as the *mode collapse* phenomenon (D'Angelo *&* Fortuin, 2021). This refers to the situation in which the SVGD particles collapse onto a single mode of the target distribution, as depicted in Figure 3.1a. When the particles experience mode collapse, the variance of the particles drastically underestimates the variance of the target distribution, in which case the particles fail to explain the uncertainty in the target distribution (Ba *et al.*, 2022). This phenomenon is analogous to the problem of *particle degeneracy* in particle filters, which refers to the situation in which only a few particles are assigned a non-negligible weight and the remaining particles have weights close to zero, and hence are redundant (e.g., Li *et al.*, 2014; Fan *et al.*, 2021).

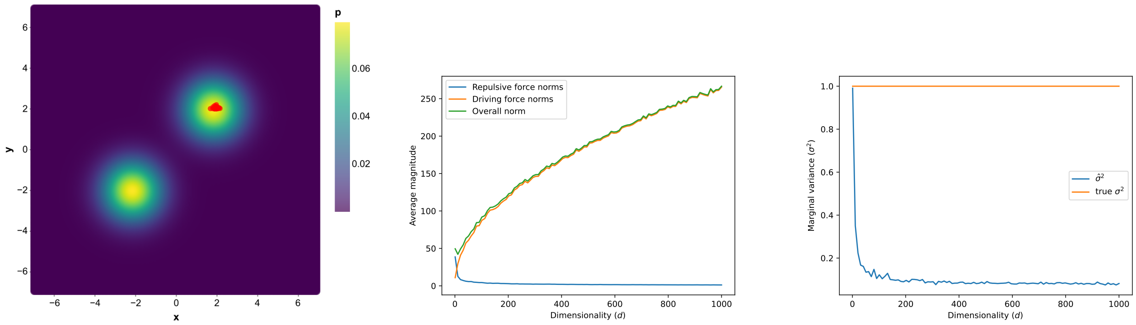
This phenomenon has been studied analytically by Zhuo *et al.* (2017) and Ba *et al.* (2022) who show that the variance/mode collapse becomes more severe as the dimension d of the target distribution increases (keeping the number of particles fixed). To understand why this is the case, let $D(x_i) = \mathbb{E}_{X_j \sim Q}\left[k(X_j, x_i)\nabla_{X_j} \log p(X_j)\right]$ and $R(x_i) = \mathbb{E}_{X_j \sim Q}\left[\nabla_{X_j} k(X_j, x_i)\right]$ respectively denote the *driving force* and *repulsive force* on x_i in the SVGD update rule. Zhuo *et al.* (2017) show that there is a negative correlation between the dimensionality d and the magnitude of the repulsive force $\|R(x_i)\|$, which leads to the mode/variance collapse in high dimensions [3]. This is a consequence of the fact that distance metrics and kernels defined in terms of a distance metric (e.g., the RBF kernel) suffer from the *curse of dimensionality* (COD) (Spigler *et al.*, 2020; Ting *et al.*, 2021), which means that the kernel similarities $k(x_j, x_i)$, and hence the gradients $\nabla_{x_j} k(x_j, x_i)$, tend to zero as the dimensionality increases. Consequently, as the dimensionality increases, the magnitude

[3] As pointed out by D'Angelo *&* Fortuin (2021), the mode/variance collapse phenomenon may also be the result of the *mode-seeking* limitation inherent to methods based on minimising the reverse KL divergence. This refers to the fact that minimisation of the reverse KL divergence leads to mode-seeking behaviour, which results in a tendency to underestimate the variance of the posterior. See Chan *et al.* (2022) for an excellent discussion on the mode-seeking limitation.

of the repulsive force $\|R(x_i)\|$ decreases dramatically, as illustrated in Figure 3.1b. This results in the SVGD dynamics becoming more dependent on the *driving force* term $D(x_i)$ (Ba *et al.*, 2022), essentially reducing SVGD to a gradient ascent algorithm for maximising the log-likelihood under the target distribution (Liu *et al.*, 2022). This is also illustrated in Figure 3.1b where the magnitudes of the driving forces become closer and closer to the overall update magnitude as the dimensionality increases.

We illustrate the variance collapse phenomenon by using SVGD to sample from a d-variate isotropic Gaussian distribution $\mathcal{N}_d(0, \sigma^2 I)$, with $\sigma^2 = 1$, as was done by Ba *et al.* (2022). We then use the SVGD particles to estimate the variance term σ^2 for increasing dimensions. As illustrated in Figure 3.1c, the variance of the target distribution (estimated by the particles) quickly tends to zero as the dimensionality increases.

Several variants of vanilla SVGD have been proposed to overcome the variance/mode collapse phenomenon. The main direction for improving upon vanilla SVGD is to use dimension reduction to project the particles and the score function $s_p(x)$ onto lower-dimensional spaces, directly combatting the COD. Along this line, Gong *et al.* (2021) propose projecting the score function and particles onto optimal one-dimensional slices, yielding a variant of SVGD called sliced-SVGD (S-SVGD). However, using one-dimensional slices is suboptimal since it results in a significant loss of information. Chen *&* Ghattas (2020) improve upon this approach by instead projecting the score function and particles onto the leading eigenvectors of some gradient information matrix, which then yields the projected SVGD (pSVGD) algorithm. However, computing the eigenvectors (and possibly the gradient information matrix itself) is computationally expensive and limits the scalability of pSVGD. A further improvement is given by Liu *et al.* (2022) who propose projecting the data onto a Grassman manifold, on which the particles are evolved according to SVGD dynamics. This approach not only effectively reduces the dimension of the problem, but also incorporates information about the underlying geometry into the updates, which is similar to the Riemann SVGD (R-SVGD) (Liu *&* Zhu, 2017) variant of SVGD. Finally, if the target distribution has a known graphical structure, message-passing variants of SVGD (e.g., Zhuo *et al.*, 2017; Zhou *&* Qiu, 2023) can alleviate the variance collapse phenomenon by identifying the Markov blanket of the graphical structure.



(a) Mode collapse.

(b) Diminishing repulsive force.

(c) Variance collapse.

Figure 3.1: Illustration of mode and variance collapse. To illustrate mode collapse (left), we use SVGD to sample from a GMM given by: $p(x) = 0.5\mathcal{N}_2(x; \begin{bmatrix}2.0 & 2.0\end{bmatrix}, I) + 0.5\mathcal{N}_2(x; \begin{bmatrix}-2.0 & -2.0\end{bmatrix}, I)$, where we initialise the particles from a bivariate Gaussian centred at the positive mode of the GMM. Since it is not possible to visualise mode collapse for a high-dimensional distribution, we mimic the effect on a two-dimensional GMM by scaling the repulsive force in the SVGD updates by an amount close to zero. We visualise the diminishing repulsive forces (middle) by plotting the average repulsive and driving force magnitudes (averaged over all particles and all iterations of SVGD) for each value of d. It is evident that the average magnitude of the repulsive force drops dramatically for increasing dimensionality, and that the converse is true for the average magnitude of the driving force. We illustrate variance collapse (right) by using SVGD particles to estimate the marginal variance σ^2 of an isotropic Gaussian $\mathcal{N}_d(0, \sigma^2 I)$ for increasing dimensionality. In all cases, we use an RBF kernel with median heuristic bandwidth - see Equation 4.10.

Other Limitations

In addition to the variance collapse phenomenon, SVGD suffers from several other, albeit less severe, limitations. Firstly, vanilla SVGD can only be used for target distributions with continuous and differentiable densities. However, Han & Liu (2018) introduce a gradient-free extension of SVGD that replaces the gradient $\nabla_x \log p(x)$ with a surrogate gradient $\nabla_x \log \rho(x)$, of an arbitrary auxiliary distribution $\rho(x)$, and uses importance weights to correct the bias induced by the surrogate gradient. Furthermore, Han *et al.* (2020) introduce a variant of SVGD that works for discrete target distributions by transforming the discrete distribution into a piecewise continuous distribution and applying the gradient-free SVGD algorithm (Han & Liu, 2018) to sample from the transformed distribution.

Another limitation of vanilla SVGD is that the performance is heavily dependent on the choice of kernel function, where the optimal kernel function cannot be determined *a priori*. To mitigate this issue, Ai *et al.* (2022) propose combining multiple kernels (e.g., combining several RBF kernels with different bandwidths) and automatically adjusting the weights of each component kernel, which then yields the Multiple-Kernel SVGD (MK-SVGD) variant of SVGD. Furthermore, Wang *et al.* (2019) propose using matrix-valued kernels to incorporate geometric information into SVGD, such as information about the local curvature provided by the Hessian matrix. This approach also effectively reduces the sensitivity to the choice of kernel function.

3.4 CONCLUSION

This chapter discussed the major advantages and limitations of vanilla SVGD, shedding light on the applicability of SVGD in practice. We discussed the fact that SVGD combines benefits from both VI and MCMC methods, and helps to alleviate some of the limitations of both these alternative approaches. While SVGD demonstrates promising advantages, several major limitations were discussed that inhibit the widespread adoption of SVGD in practice. Specifically, SVGD suffers from mode/variance collapse in high dimensions and may be sensitive to the choice of kernel function. Fortunately, several extensions of vanilla SVGD have been proposed to alleviate these limitations, several of which were discussed in this chapter.

The following chapter presents an application of SVGD in reinforcement learning known as the Stein Variational Policy Gradient method, which aims at learning "a set of diverse but well-behaved policies" (Liu *et al.*, 2017).

CHAPTER 4

SVGD IN REINFORCEMENT LEARNING VIA THE STEIN VARIATIONAL POLICY GRADIENT METHOD

4.1 INTRODUCTION

Reinforcement Learning (RL) has emerged as a powerful paradigm for agents to learn to make sequential decisions by interacting with an environment. The goal of RL can be succinctly summarised as learning a policy, denoted by $\pi(a|s)$, which informs an agent of promising actions to take in a given state; usually, promise is defined in terms of maximising an expected future reward signal.

This chapter discusses a particular application of SVGD in RL known as the Stein Variational Policy Gradient (SVPG) method (Liu *et al.*, 2017). We first provide a brief background on function approximation and policy gradient methods in RL, and discuss the motivation for using SVGD in this context.

4.2 BACKGROUND AND PRELIMINARIES

This section provides a brief overview of background information relevant to SVPG.

Notations We denote the action and state space of the environment by $\mathcal{A}$ and $\mathcal{S}$, respectively. The reward function is denoted by $r(s_t, a_t)$, which specifies the numerical reward received by an agent taking action a_t in state s_t at time t.

Function Approximation

Tabular methods in RL, such as tabular Q-learning and SARSA, have been widely used in practice. However, when the action and/or state space is continuous or has a large dimensionality, tabular methods become infeasible (Sutton *&* Barto, 2017). Hence, function approximation has been introduced to model the action-value function $Q(s, a)$[1] by a parameterised function $f_\theta(s, a)$ (e.g., Long *&* Han, 2023). The parameters θ are optimised to approximate the optimal value function,

[1] In some cases, the state-value function, $V(s)$, is used instead.

$Q_*(s,a)$. This circumvents the problem of having to store extremely large tables in memory, as well as the problem of encountering states that have not yet been visited (and hence do not have value estimates stored in the table).

Policy Gradient Methods

In value-based RL methods, the aim is to learn (or approximate) the optimal value function $Q_*(s,a)$, which implicitly specifies the optimal policy by $\pi_*(a|s) = \arg\max_{a\in\mathcal{A}} Q_*(s,a)\ \forall s \in \mathcal{S}$. Conversely, policy-based methods aim to learn (or approximate) the optimal policy $\pi_*(a|s)$ directly (Mnih *et al.*, 2016). Policy gradient methods refer to policy-based function approximation methods that assume a parametric form for the policy, $\pi(a|s;\theta) \equiv \pi_\theta(a|s)$, and learn the parameters θ by performing (approximate) gradient ascent on some performance measure, $J(\pi_\theta(a|s))$ (Sutton *&* Barto, 2017). We write $J(\pi_\theta(a|s)) \equiv J(\theta)$ to simplify notation. The update rule for the parameters can now be given by:

$$\theta_{t+1} = \theta_t + \epsilon_t \widehat{\nabla_\theta J(\theta_t)}\big|_{\theta=\theta_t} \tag{4.1}$$

where $\widehat{\nabla_\theta J(\theta_t)}$ is an estimate of the true gradient, $\nabla_\theta J(\theta_t)$, and ϵ_t is a step size. In the episodic case, the performance measure is given by the value of the start state of the episode (Sutton *&* Barto, 2017):

$$J(\theta) = V_{\pi_\theta}(s_0) = \sum_{a\in\mathcal{A}} \pi(a|s_0) Q_{\pi_\theta}(s_0,a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t)\right] \tag{4.2}$$

The *policy gradient theorem* (Sutton *et al.*, 1999) provides a closed-form expression for the gradient of the performance measure. In the episodic case, the gradient of Equation (4.2) is given by:

$$\nabla_\theta J(\theta) = \frac{1}{Z} \sum_{s\in\mathcal{S}} d_{\pi_\theta}(s) \sum_{a\in\mathcal{A}} Q_{\pi_\theta}(s,a) \nabla_\theta \pi_\theta(a|s) \tag{4.3}$$

where d_{π_θ} denotes the stationary distribution of the Markov chain for π_θ and Z is a constant of proportionality that is equal to the average length of an episode in the episodic case, and is equal to one in the continuing case (Sutton *&* Barto, 2017) [2].

In practice, the policy gradient in Equation (4.3) can be estimated using either finite difference methods or likelihood ratio-based methods (e.g., Peters, 2010; Liu *et al.*, 2017). Two popular

[2] In practice, the constant of proportionality, Z, is not important as it can be absorbed into the step size.

examples of likelihood ratio-based policy gradient methods are the REINFORCE method (Williams, 2004) and Actor-Critic methods[3]. The REINFORCE policy gradient estimator for a single rollout trajectory is given by Liu *et al.* (2017):

$$\nabla_\theta J(\theta) = \sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t|s_t) G_t$$

where $G_t = \sum_{i=0}^{\infty} \gamma^i r(s_{t+1}, a_{t+i})$ is the discounted cumulative return.

Motivation for SVPG

Although policy gradient methods have proven useful in practice, they suffer from high variance and insufficient exploration (Liu *et al.*, 2017; Cohen *et al.*, 2019). This is problematic since exploration is often key to the success of RL algorithms, especially in the context of sparse rewards (see, Ladosz *et al.*, 2022). Liu *et al.* (2017) propose using the SVGD algorithm (see Algorithm 1) to improve the exploration of policy gradient methods by training a set of diverse policies $\{\pi_{\theta_i}(a|s)\}_{i=1}^{n}$. In this approach, the repulsive force term in SVGD - see Equation (2.25) - encourages diversity among the particles, which yields improved exploration in parameter space.

4.3 STEIN VARIATIONAL POLICY GRADIENT

SVPG is a policy gradient method that builds on the concept of maximum-entropy RL (Liu *et al.*, 2017) and is fundamentally different from traditional policy gradient methods. In traditional policy gradient methods, the parameters θ of the policy $\pi_\theta(a|s)$ are treated as fixed parameters and are optimised to maximise the expected return $J(\theta)$. In contrast, SVPG treats the parameters θ as a random variable and attempts to approximate the distribution Q, with density function $q(\theta)$, that maximises the entropy-regularised expected return (Liu *et al.*, 2017) given by:

$$\tilde{J}(q) = \max_q \left\{ \mathbb{E}_{\theta \sim Q} \left[J(\theta) \right] + \alpha \mathbb{H}(q) \right\} \tag{4.4}$$

where $\mathbb{H}(q)$ denotes the entropy of the distribution $q(\theta)$ and α can be viewed as a temperature parameter (Liu *et al.*, 2017). This entropy-regularised optimisation problem "explicitly encourages

[3] For a detailed overview and discussion of policy gradient methods, see the blog post by Weng (2018).

exploration in the θ parameter space according to the principle of maximum entropy"[4] (Liu *et al.*, 2017). Furthermore, this framework allows one to include prior knowledge in the form of a prior distribution $q_0(\theta)$, which can also be used to provide regularisation (Liu *et al.*, 2017). In this case, the entropy term $\mathbb{H}(q)$ in Equation (4.4) is replaced by the (reverse) KL divergence between the distribution $q(\theta)$ and the prior $q_0(\theta)$, yielding the optimisation problem given by Liu *et al.* (2017):

$$\tilde{J}(q) = \max_{q} \left\{ \mathbb{E}_{\theta \sim Q}\left[J(\theta)\right] - \alpha \mathbb{D}_{\mathrm{KL}}(q||q_0) \right\} \quad . \tag{4.5}$$

The optimal distribution, $q^*(\theta)$, of the above optimisation problem can be derived by setting $\nabla_q \tilde{J}(q) = 0$ and solving for q (Liu *et al.*, 2017), which yields (see Appendix A for the derivation):

$$q^*(\theta) \propto \exp\left(\frac{1}{\alpha} J(\theta)\right) q_0(\theta) \quad . \tag{4.6}$$

As discussed by Liu *et al.* (2017), the optimal distribution $q^*(\theta)$ above can be viewed as the posterior distribution of the parameters θ given the likelihood, $\exp\left(\frac{1}{\alpha}J(\theta)\right)$, and prior $q_0(\theta)$. Given this posterior distribution, the SVGD algorithm presented in Chapter 2 can be applied directly to approximate this posterior. That is, one initialises a set of particles $\{\theta_i\}_{i=1}^n$, where each particle represents the parameters of a policy function approximation, $\pi_{\theta_i}(a|s)$, and iteratively updates the positions of the particles using the SVGD update equation given by Liu *et al.* (2017):

$$\theta_i \leftarrow \theta_i + \frac{\epsilon}{n} \sum_{j=1}^{n} \underbrace{k(\theta_j, \theta_i) \nabla_{\theta_j} \left(\frac{1}{\alpha} J(\theta_j) + \log q_0(\theta_j)\right)}_{\text{exploitation}} + \underbrace{\nabla_{\theta_j} k(\theta_j, \theta_i)}_{\text{exploration}} \tag{4.7}$$

In the update rule in Equation (4.7), the gradient $\nabla_\theta J(\theta)$ can be computed using any existing policy gradient method such as REINFORCE (Williams, 2004) or the Actor-Critic method along with any of its variants (Liu *et al.*, 2017).

Remark: The **driving force** in Equation (2.25) now serves as an **exploitation** term in Equation (4.7), and the **repulsive force** in Equation (2.25) now serves as an **exploration** term in Equation (4.7) (Liu *et al.*, 2017). Furthermore, the exploration-exploitation trade-off in the SVPG algorithm is controlled by the temperature parameter α (Liu *et al.*, 2017).

[4] For a discussion and analysis of the principle of maximum entropy in RL, see Eysenbach & Levine (2022).

Remark: The SVPG algorithm contains three sources of exploration: the first source of exploration arises from the prior regularisation of $q_0(\theta)$, which encourages exploration by the principle of maximum entropy (Liu *et al.*, 2017); the second source of exploration is the deterministic repulsive force of the SVGD algorithm, which encourages diversity among the particles $\{\theta_i\}_{i=1}^n$; the third source of exploration is quite subtle, and arises from the use of softmax action selection (assuming a discrete action space) when simulating episodes for training. The use of softmax action selection encourages exploration similar to that of ϵ-greedy action-selection approaches.

Variance collapse of SVPG

As discussed in Chapter 3, the major limitation of SVGD is the so-called mode/variance collapse phenomenon. In SVPG, each particle θ_i represents the parameters of (typically) a neural network policy $\pi_{\theta_i}(a|s)$, which may be very high dimensional. Hence, it may seem reasonable to suspect that the mode/variance collapse phenomenon would be exacerbated in SVPG. However, since the target distribution is non-stationary and changes as more information is gained by the agent(s), the modes of the target distribution are also non-stationary. Hence, SVPG may be less prone to mode/variance collapse than SVGD.

Furthermore, it is important to note that the repulsive force in SVPG serves a distinct purpose compared to SVGD. In SVGD, the ultimate goal is to obtain a representative sample from the posterior distribution, and the repulsive force is used to encourage diversity in the particle positions. In SVPG, the ultimate goal is to obtain an accurate approximation of the optimal policy, $\pi_*(a|s)$, and the repulsive force is used to encourage exploration of the environment. Therefore, it is not cause for concern if the particles experience mode collapse during the later stages of training, so long as the particles adequately explored the environment before collapsing to a mode, and hence are able to collapse to a (near-)optimal mode.

However, mode collapse in the early stages of training is problematic. Liu *et al.* (2017) discuss that when the temperature parameter is very small ($\alpha \to 0$), which is nearly equivalent to a zero repulsive force magnitude, the SVPG algorithm essentially reduces to running n independent policy gradient algorithms for each of the policies $\pi_{\theta_i}(a|s)$. Therefore, it is imperative to ensure that the repulsive force is sufficiently strong during the early stages of training. This can, to some extent, be achieved by carefully annealing the temperature parameter to ensure that the repulsive force dominates during the early stages of training and the driving force dominates during later stages, thereby exploiting the information gained in the early stages of training (Liu *et al.*, 2017).

A Novel Variant of Vanilla SVPG

We propose a simple, novel variant of SVPG to improve upon vanilla SVPG in terms of exploration and sensitivity to the choice of kernel function.

Specifically, we leverage the idea of using a linear combination of multiple kernels instead of using a single kernel, as was done by Ai *et al.* (2022) for SVGD. Ai *et al.* (2022) propose using a linear combination of RBF kernels, each having a fixed (at the start of training) bandwidth. Instead, we propose using a linear combination of RBF kernels, each having a bandwidth proportional to the median heuristic bandwidth, σ_{med}, given by:

$$\sigma_{\text{med}} = \sqrt{\frac{med^2}{2\log(n+1)}} \tag{4.10}$$

where *med* denotes the median pairwise distance between particles. That is, we consider using a set of component kernels $\{k_l(\theta, \theta')\}_{l=1}^m$, each having a bandwidth proportional to the median heuristic bandwidth, $\sigma_l \propto \sigma_{\text{med}}$. In this way, the bandwidth of each component kernel adapts to the spread of the current particles. The mixture kernel is then given by:

$$k_{\text{mix}}(\theta, \theta') = \sum_{l=1}^{m} w_l k_l(\theta, \theta') \tag{4.11}$$

where $w_l \in (0, 1)$ denotes the weight of the l^{th} component kernel such that $\sum_{l=1}^m w_l = 1$. Given this mixture kernel, the optimal update direction given in Equation (2.25) may be computed for each component kernel as follows:

$$\phi_l^*(\cdot) = \frac{1}{n}\sum_{j=1}^{n} k_l(\theta_j, \cdot)\nabla_{\theta_j}\left[\frac{1}{\alpha}\cdot J(\theta_j) + \log q_0(\theta_j)\right] + \nabla_{\theta_j} k_l(\theta_j, \cdot) \ . \tag{4.12}$$

Given the optimal update direction for each component kernel, $\phi_l^*(\cdot)$, the optimal update direction can be computed as a linear combination of these optimal directions given by:

$$\phi^*_{\text{mix}}(\cdot) = \sum_{l=1}^{m} w_l \phi^*_l(\cdot) \tag{4.13}$$

where the weights are given by: [5]

$$w_l = \frac{\|\phi^*_l(\cdot)\|_{\mathcal{H}}}{\sum_{j=1}^{m} \|\phi^*_j(\cdot)\|_{\mathcal{H}}} \tag{4.14}$$

where $\|\cdot\|_{\mathcal{H}}$ denotes the norm induced by the inner product, $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, of the RKHS $\mathcal{H}$ corresponding to the positive definite kernel $k_{\text{mix}}(\cdot, \cdot)$.

The resulting Mixture-Kernel Stein Variational Policy Gradient (MK-SVPG) algorithm is summarised in Algorithm 2.

4.4 EXPERIMENTS

In this section, we conduct several experiments to illustrate the effectiveness of SVPG. Specifically, we consider two classic control problems in the gym package (Brockman *et al.*, 2016): "Cartpole-v1" and "Acrobot-v1". Furthermore, we also consider the "LunarLander-v2" Box2D environment in the gym package (Brockman *et al.*, 2016). For each experiment, we implement vanilla SVPG and our variant, MK-SVPG, using the REINFORCE method to calculate policy gradients. Furthermore, we use the REINFORCE algorithm to serve as a baseline for comparison. To allow a fair comparison, we adopt the "REINFORCE-Independent" method (Liu *et al.*, 2017) wherein multiple agents are trained independently, each using the original REINFORCE method.

In all cases, we use $n = 16$ policies, a discount rate of $\gamma = 0.99$, and use ADAM (Kingma *&* Ba, 2014) to set step sizes for gradient descent updates (with initial learning rates of 0.01, 0.001, and 0.001 for CartPole, Acrobot, and LunarLander respectively). Furthermore, for each of the experiments, the policies are parameterised by a neural network with a single hidden layer containing 128 neurons and ReLU activation. For SVPG and MK-SVPG, we follow Liu *et al.* (2017) by using an initial temperature of $\alpha_0 = 10$ and using a flat improper prior given by $\log q_0(\theta) = 1$ in all experiments. Furthermore, in each case, we use a simple geometric cooling schedule for the temperature parameter given by $\alpha_t = \alpha_0(1 - \delta)^t$. For CartPole, we train the agents for 50 episodes using 20 rollout

[5] In our practical experiments, we instead use the Euclidean norm for simplicity.

trajectories to calculate the policy gradients. For Acrobot, we train the agents for 100 episodes using 5 rollout trajectories to calculate the policy gradients. For LunarLander, we train the agents for 500 episodes using 10 rollout trajectories to calculate the policy gradients.

After training, we evaluate the policies for each algorithm in three ways: firstly, we evaluate each of the policies independently and report the average performance across policies; secondly, we use the first evaluation method to determine the best-performing policy, and evaluate the performance of this policy; thirdly, we evaluate a naive Bayes ensemble policy wherein actions are selected according to the highest estimated probability across policies.

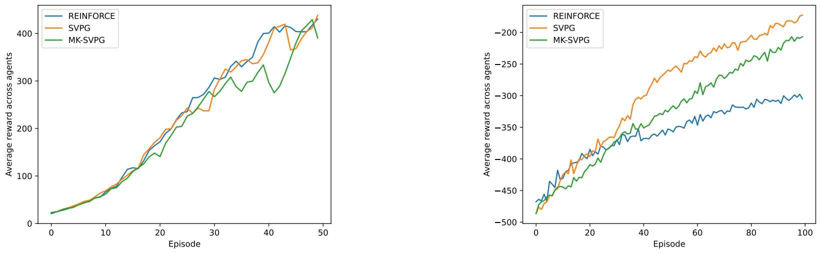
The learning curves for the three algorithms on the classic control experiments are given in Figure 4.1, and the average rewards over 100 evaluation episodes are summarised in Table 4.1. The learning curves on the LunarLander environment are given in Figure 4.2, and the average rewards over 100 evaluation episodes are summarised in Table 4.2.

For the classic control problems, all three algorithms are able to solve the environments. For the CartPole environment, it seems that all three algorithms yield similar performance, as evident in Figure 4.1a. However, from Figure 4.1b it is appears that both SVPG and MK-SVPG learn much faster than REINFORCE on the Acrobot problem. For the LunarLander environment, it is evident from Figure 4.2 and Table 4.2 that both SVPG and MK-SVPG are able to solve the environment, whereas REINFORCE cannot. This is likely due to the LunarLander environment having a much sparser reward structure than the classic control environments. The results suggest that, while MK-SVPG does yield better exploration compared to SVPG, it is not clear whether MK-SVPG yields better performance on these tasks. More experiments are needed to determine the relative performance of these two algorithms, especially on more complex environments where exploration is crucial.

	Rewards		
Algorithm	Average	Best	Ensemble
REINFORCE	476.51	500.00	500.00
SVPG	473.17	500.00	500.00
MK-SVPG	415.23	500.00	500.00

	Rewards		
Algorithm	Average	Best	Ensemble
REINFORCE	-320.38	-81.05	-500.00
SVPG	-196.62	-84.47	-455.73
MK-SVPG	-291.71	-79.12	-86.23

Table 4.1: Average rewards over 100 evaluation episodes for CartPole (left) and Acrobot (right).



(a) CartPole.

(b) Acrobot.

Figure 4.1: Learning curves for the three algorithms on classic control problems: CartPole (left) and Acrobot (right).

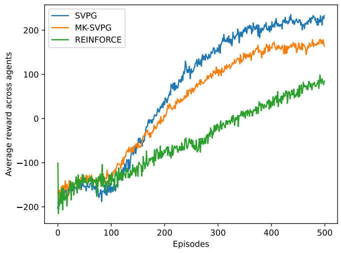


Figure 4.2: Learning curves for LunarLander.

	Rewards		
Algorithm	Average	Best	Ensemble
REINFORCE	-45.94	80.70	-27.70
SVPG	182.78	250.50	231.43
MK-SVPG	97.73	247.80	241.03

Table 4.2: Average rewards over 100 evaluation episodes for LunarLander.

4.5 CONCLUSION

This chapter presented the Stein Variational Policy Gradient (SVPG) (Liu *et al.*, 2017) method and introduced a novel variant thereof, called Multiple-Kernel SVPG, that uses a linear combination of RBF kernels. Furthermore, the effectiveness of both SVPG and MK-SVPG was demonstrated on two classic control problems, CartPole and Acrobot, and one Box2D environment, LunarLander, from the gym package (Brockman *et al.*, 2016). The results demonstrate that SVPG and MK-SVPG yield significant improvement over REINFORCE. While our MK-SVPG variant shows promise, more experiments are needed to conclusively determine its performance relative to vanilla SVPG, especially on more complex environments where significant exploration is required.

CHAPTER 5

CONCLUSION

5.1 SUMMARY

This report provided a comprehensive survey of Stein Variational Gradient Descent (SVGD). Chapter 2 presented the statistical development of SVGD and demonstrated its effectiveness on a simple sampling problem, where SVGD was able to outperform three well-known MCMC sampling algorithms. Chapter 3 discussed the major advantages and limitations of SVGD, and discussed several extensions of vanilla SVGD that may alleviate some of its major limitations such as the variance/mode collapse phenomenon and sensitivity to choice of kernel function. Chapter 4 presented the Stein Variational Policy Gradient (SVPG) method, together with a novel variant thereof called the Multiple-Kernel Stein Variational Policy Gradient (MK-SVPG) method. Experiments were conducted on several gym problems which demonstrated the effectiveness of SVPG and MK-SVPG compared to REINFORCE.

5.2 CONCLUSION

In conclusion, SVGD shows significant promise as a general-purpose tool for accurate approximation of complex target distributions. The non-parametric nature of SVGD makes it very flexible and amenable to a wide range of challenging tasks, and potentially allows SVGD to yield more accurate approximations than traditional variational inference methods. Furthermore, the deterministic, gradient-based updates in SVGD potentially allows SVGD to converge much faster than Markov chain Monte Carlo methods.

SVPG also shows significant promise in reinforcement learning for solving complex problems where exploration is critical. The use of entropy-regularisation and a kernel repulsive force allows SVPG to more effectively explore the environment compared to traditional policy gradient methods.

5.3 FUTURE WORK

The main direction for future research in SVGD is to analyse the convergence properties and establish explicit convergence rates in the finite-particle and finite-time regime.

There are two main directions of future research for SVPG: firstly, a more comprehensive comparative analysis of SVPG to other existing methods on complex tasks is needed; secondly, it may be valuable to investigate using existing extensions of SVGD in the SVPG method.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. & Zheng, X. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
Available at: https://www.tensorflow.org/

Ai, Q., Liu, S., He, L. & Xu, Z. 2022. Stein variational gradient descent with multiple kernels. *Cognitive Computation*, 15(2):672–682.
Available at: https://doi.org/10.1007%2Fs12559-022-10069-5

Anastasiou, A., Barp, A., Briol, F.-X., Ebner, B., Gaunt, R.E., Ghaderinezhad, F., Gorham, J., Gretton, A., Ley, C., Liu, Q., Mackey, L., Oates, C.J., Reinert, G. & Swan, Y. 2023. Stein's Method Meets Computational Statistics: A Review of Some Recent Developments. *Statistical Science*, 38(1):120 – 139.
Available at: https://doi.org/10.1214/22-STS863

Ba, J., Erdogdu, M.A., Ghassemi, M., Sun, S., Suzuki, T., Wu, D. & Zhang, T. 2022. Understanding the variance collapse of SVGD in high dimensions. In *International Conference on Learning Representations*.
Available at: https://openreview.net/forum?id=Qycd9j5Qp9J

Barbour, A.D. 1990. Stein's method for diffusion approximations. *Probability Theory and Related Fields*, 84(3):297–322. ISSN 1432-2064.
Available at: https://doi.org/10.1007/BF01197887

Barp, A., Briol, F.-X., Duncan, A.B., Girolami, M. & Mackey, L. 2022. Minimum Stein Discrepancy Estimators. *Proceedings of the 33rd Conference on Neural Information Processing Systems.*

Blei, D.M., Kucukelbir, A. & McAuliffe, J.D. 2017. Variational inference: A review for statisticians.

Journal of the American Statistical Association, 112(518):859–877.
Available at: https://doi.org/10.1080%2F01621459.2017.1285773

Bonis, T. 2020. Stein's method for normal approximation in Wasserstein distances with application to the multivariate central limit theorem. *Probability Theory and Related Fields*, 178.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J. & Zaremba, W. 2016. Openai gym.

Budhiraja, A., Dupuis, P. & Fischer, M. 2012. Large deviation properties of weakly interacting processes via weak convergence methods. *The Annals of Probability*, 40(1).
Available at: https://doi.org/10.1214%2F10-aop616

Chan, A., Silva, H., Lim, S., Kozuno, T., Mahmood, A.R. & White, M. 2022. Greedification Operators for Policy Optimization: Investigating Forward and Reverse KL Divergences. *J. Mach. Learn. Res.*, 23(1). ISSN 1532-4435.

Chatterjee, S. 2014. A short survey of Stein's method. *arXiv: Probability*.
Available at: https://api.semanticscholar.org/CorpusID:118347930

Chen, L.H.Y. 2021. Stein's method of normal approximation: Some recollections and reflections. *The Annals of Statistics*, 49(4):1850–1863. © Institute of Mathematical Statistics, 2021.

Chen, P. & Ghattas, O. 2020. Projected stein variational gradient descent. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan & H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pages 1947–1958. Curran Associates, Inc.
Available at: https://proceedings.neurips.cc/paper_files/paper/2020/file/14faf969228fc18fcd4fcf59437b0c97-Paper.pdf

Chwialkowski, K., Strathmann, H. & Gretton, A. 2016. A kernel test of goodness of fit. *Proceedings of The 33rd International Conference on Machine Learning*, 48:2606–2615.
Available at: https://proceedings.mlr.press/v48/chwialkowski16.html

Cohen, A., Qiao, X., Yu, L., Way, E. & Tong, X. 2019. Diverse exploration via conjugate policies for policy gradient methods. *Proceedings of the AAAI Conference on Artificial Intelligence*,

33(01):3404–3411.
Available at: https://ojs.aaai.org/index.php/AAAI/article/view/4215

D'Angelo, F. & Fortuin, V. 2021. Annealed Stein Variational Gradient Descent. *Proceedings of the 3rd Symposium on Advances in Approximate Bayesian Inference*, pages 1–12.

Das, A. & Nagaraj, D. 2023. Provably Fast Finite Particle Variants of SVGD via Virtual Particle Stochastic Approximation. *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023).*

Detommaso, G., Cui, T., Spantini, A., Marzouk, Y. & Scheichl, R. 2018. A Stein variational Newton method. *Advances in Neural Information Processing Systems (NIPS) 2018.*

Duane, S., Kennedy, A., Pendleton, B.J. & Roweth, D. 1987. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222. ISSN 0370-2693.
Available at: https://www.sciencedirect.com/science/article/pii/037026938791197X

Eysenbach, B. & Levine, S. 2022. Maximum Entropy RL (Provably) Solves Some Robust RL Problems. *International Conference on Learning Representations.*
Available at: https://openreview.net/forum?id=PtSAD3caaA2

Fan, J., Taghvaei, A. & Chen, Y. 2021. Stein particle filtering. *CoRR*, abs/2106.10568.
Available at: https://arxiv.org/abs/2106.10568

Ganguly, A. & Earp, S.W.F. 2021. An introduction to variational inference. *ArXiv*, abs/2108.13083.
Available at: https://api.semanticscholar.org/CorpusID:237353445

Gaunt, R.E., Mijoule, G. & Swan, Y. 2019. An algebra of stein operators. *Journal of Mathematical Analysis and Applications*, 469(1):260–279. ISSN 0022-247X.
Available at: https://www.sciencedirect.com/science/article/pii/S0022247X1830756X

Ghojogh, B., Ghodsi, A., Karray, F. & Crowley, M. 2021. Reproducing Kernel Hilbert Space, Mercer's Theorem, Eigenfunctions, Nyström Method, and Use of Kernels in Machine Learning: Tutorial and Survey. *ArXiv*, abs/2106.08443.
Available at: https://api.semanticscholar.org/CorpusID:235446387

Gong, W., Li, Y. & Hernández-Lobato, J.M. 2021. Sliced Kernelized Stein Discrepancy. In *International Conference on Learning Representations (ICLR)*. Published as a conference paper at ICLR 2021.

Available at: https://openreview.net/forum?id=1jDFcof5P9v

Gorham, J., Duncan, A.B., Vollmer, S.J. & Mackey, L. 2019. Measuring sample quality with diffusions. *The Annals of Applied Probability*, 29(5):2884–2928. © Institute of Mathematical Statistics, 2019.

Gorham, J. & Mackey, L. 2015. Measuring Sample Quality with Stein's Method. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 226–234. Cambridge, MA, USA: MIT Press.

Gorham, J., Raj, A. & Mackey, L. 2020. Stochastic stein discrepancies. In *Advances in Neural Information Processing Systems*, volume 33, pages 17931–17942. Curran Associates, Inc.

Available at: https://proceedings.neurips.cc/paper_files/paper/2020/file/d03a857a23b5285736c4d55e0bb067c8-Paper.pdf

Gunapati, G., Jain, A., Srijith, P.K. & Desai, S. 2022. Variational inference as an alternative to MCMC for parameter estimation and model selection. *Publications of the Astronomical Society of Australia*, 39.

Available at: https://doi.org/10.1017%2Fpasa.2021.64

Han, J., Ding, F., Liu, X., Torresani, L., Peng, J. & Liu, Q. 2020. Stein variational inference for discrete distributions. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4563–4572. PMLR.

Available at: https://proceedings.mlr.press/v108/han20c.html

Han, J. & Liu, Q. 2018. Stein variational gradient descent without gradient. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1900–1908. PMLR.

Available at: https://proceedings.mlr.press/v80/han18b.html

Hoeffding, W. 1948. A Class of Statistics with Asymptotically Normal Distribution. *The Annals of Mathematical Statistics*, 19(3):293 – 325.
Available at: https://doi.org/10.1214/aoms/1177730196

Hoffman, M. & Gelman, A. 2011. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15.

Hu, T., Chen, Z., Sun, H., Bai, J., Ye, M. & Cheng, G. 2021. Stein Neural Sampler. *ArXiv*, abs/1810.03545.
Available at: https://api.semanticscholar.org/CorpusID:52938806

Hyvärinen, A. 2005. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709.
Available at: http://jmlr.org/papers/v6/hyvarinen05a.html

Kanagawa, H., Jitkrittum, W., Mackey, L., Fukumizu, K. & Gretton, A. 2023. A kernel Stein test for comparing latent variable models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(3):986–1011.
Available at: https://doi.org/10.1093%2Fjrsssb%2Fqkad050

Kingma, D.P. & Ba, J. 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.
Available at: https://api.semanticscholar.org/CorpusID:6628106

Ladosz, P., Weng, L., Kim, M. & Oh, H. 2022. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22.
Available at: https://doi.org/10.1016%2Fj.inffus.2022.03.003

Ley, C., Reinert, G. & Swan, Y. 2014. Approximate computation of expectations: the canonical stein operator. *Research Papers in Economics*.
Available at: https://api.semanticscholar.org/CorpusID:118846588

Ley, C., Reinert, G. & Swan, Y. 2017. Stein's method for comparison of univariate distributions. *Probability Surveys*, 14:1–52. ISSN 1549-5787.

Ley, C. & Swan, Y. 2013. Stein's density approach and information inequalities. *Electronic Communications in Probability*, 18(7):1–14. ISSN 1083-589X.

Li, T., Sun, S., Sattar, T.P. & Corchado, J.M. 2014. Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches. *Expert Systems with Applications*, 41(8):3944–3954.
Available at: https://doi.org/10.1016%2Fj.eswa.2013.12.031

Liu, C. & Zhu, J. 2017. Riemannian Stein Variational Gradient Descent for Bayesian Inference. In *AAAI Conference on Artificial Intelligence*.
Available at: https://api.semanticscholar.org/CorpusID:19150312

Liu, Q. 2016. Stein variational gradient descent: Theory and applications.
Available at: https://api.semanticscholar.org/CorpusID:14946786

Liu, Q. 2017. Stein Variational Gradient Descent as Gradient Flow. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
Available at: https://proceedings.neurips.cc/paper_files/paper/2017/file/17ed8abedc255908be746d245e50263a-Paper.pdf

Liu, Q. & Feng, Y. 2017. Two methods for wild variational inference.
Available at: https://openreview.net/forum?id=Sy4tzwqxe

Liu, Q., Lee, J. & Jordan, M. 2016. A Kernelized Stein Discrepancy for Goodness-of-fit Tests. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 276–284. New York, New York, USA: PMLR.
Available at: https://proceedings.mlr.press/v48/liub16.html

Liu, Q. & Wang, D. 2016. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016)*. Barcelona, Spain.
Available at: https://arxiv.org/abs/1608.04471

Liu, Q. & Wang, D. 2018. Stein variational gradient descent as moment matching. *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*.

Liu, X., Zhu, H., Ton, J.-F., Wynne, G. & Duncan, A. 2022. Grassmann Stein Variational Gradient Descent. In *Proceedings of The 25th International Conference on Artificial Intelligence and*

Statistics, volume 151 of *Proceedings of Machine Learning Research*, pages 2002–2021. PMLR. Available at: https://proceedings.mlr.press/v151/liu22a.html

Liu, Y., Ramachandran, P., Liu, Q. & Peng, J. 2017. Stein Variational Policy Gradient. *ArXiv*, abs/1704.02399.
Available at: https://api.semanticscholar.org/CorpusID:4410100

Long, J. & Han, J. 2023. Reinforcement Learning with Function Approximation: From Linear to Nonlinear. *Journal of Machine Learning*, 2(3):161–193. ISSN 2790-2048.
Available at: http://global-sci.org/intro/article_detail/jml/22011.html

Lu, J., Lu, Y. & Nolen, J. 2019. Scaling Limit of the Stein Variational Gradient Descent: The Mean Field Regime. *SIAM Journal on Mathematical Analysis*, 51(2):648–671.
Available at: https://doi.org/10.1137/18M1187611

Lunde, R. 2019. Sample splitting and weak assumption inference for time series. *arXiv preprint arXiv:1902.07425.*

Mijoule, G., Raič, M., Reinert, G. & Swan, Y. 2023. Stein's density method for multivariate continuous distributions. *Electronic Journal of Probability*, 28(none):1 – 40.
Available at: https://doi.org/10.1214/22-EJP883

Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D. & Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937. New York, New York, USA: PMLR.
Available at: https://proceedings.mlr.press/v48/mniha16.html

Müller, A. 1997. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443. ISSN 00018678.
Available at: http://www.jstor.org/stable/1428011

Oates, C.J., Girolami, M. & Chopin, N. 2017. Control functionals for Monte Carlo integration. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 79(3):695–718. ISSN 13697412, 14679868.
Available at: http://www.jstor.org/stable/44681807

Papamakarios, G., Nalisnick, E., Rezende, D.J., Mohamed, S. & Lakshminarayanan, B. 2021. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64.
Available at: http://jmlr.org/papers/v22/19-1028.html

Peters, J. 2010. Policy gradient methods. *Scholarpedia*, 5:3698.

Pinder, T., Nemeth, C. & Leslie, D. 2020. Stein Variational Gaussian Processes. *ArXiv*, abs/2009.12141.
Available at: https://api.semanticscholar.org/CorpusID:221949365

Pinski, F.J., Simpson, G., Stuart, A.M. & Weber, H. 2015. Kullback–leibler approximation for probability measures on infinite dimensional spaces. *SIAM Journal of Mathematical Analysis*, 47(6). ISSN 0036-1410.
Available at: https://www.osti.gov/biblio/1459163

Pu, Y., Gan, Z., Henao, R., Li, C., Han, S. & Carin, L. 2017. Vae learning via stein variational gradient descent. *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*.

Reinert, G. 1998. Couplings for normal approximations with stein's method. In D. Aldous & J. Propp (eds.), *Microsurveys in Discrete Probability*, Dimacs series, pages 193–207. AMS.
Available at: https://www.stats.ox.ac.uk/~reinert/papers/steinrevrev.pdf

Rice, J.A. 2007. *Mathematical Statistics and Data Analysis.* 3rd edition. Cengage Learning.

Robert, C., Elvira, V., Tawn, N. & Wu, C. 2018. Accelerating MCMC algorithms. *Wiley Interdisciplinary Reviews: Computational Statistics*, 10.

Ross, N. 2011. Fundamentals of stein's method. *Probability Surveys*, 8:210–293. ISSN 1549-5787.

Salim, A., Sun, L. & Richtarik, P. 2022. A Convergence Theory for SVGD in the Population Limit under Talagrand's Inequality T1. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 19139–19152. PMLR.
Available at: https://proceedings.mlr.press/v162/salim22a.html

Salimans, T., Kingma, D.P. & Welling, M. 2015. Markov chain monte carlo and variational inference: Bridging the gap. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *JMLR: W&CP*. Lille, France.

Shi, J. & Mackey, L.W. 2022. A Finite-Particle Convergence Rate for Stein Variational Gradient Descent. *ArXiv*, abs/2211.09721.
Available at: https://api.semanticscholar.org/CorpusID:253581833

South, L.F., Riabiz, M., Teymur, O. & Oates, C.J. 2022. Postprocessing of MCMC. *Annual Review of Statistics and Its Application*, 9(1):529–555.
Available at: https://doi.org/10.1146%2Fannurev-statistics-040220-091727

Spigler, S., Geiger, M. & Wyart, M. 2020. Asymptotic learning curves of kernel methods: empirical data versus teacher–student paradigm. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124001.
Available at: https://doi.org/10.1088%2F1742-5468%2Fabc61d

Sriperumbudur, B.K., Fukumizu, K., Gretton, A., Scholkopf, B. & Lanckriet, G.R.G. 2009. On integral probability metrics, ϕ-divergences and binary classification. *arXiv: Information Theory*.
Available at: https://api.semanticscholar.org/CorpusID:14114329

Stein, C. 1972. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability*, pages 583–602.
Available at: https://api.semanticscholar.org/CorpusID:53492374

Stein, C., Diaconis, P., Holmes, S. & Reinert, G. 2004. Use of exchangeable pairs in the analysis of simulations. *Lecture Notes-Monograph Series*, 46:1–26. ISSN 07492170.
Available at: http://www.jstor.org/stable/4356331

Sutton, R.S. & Barto, A.G. 2017. *Reinforcement Learning: An Introduction*. A Bradford Book, 2nd edition. Cambridge, Massachusetts, London, England: The MIT Press. In progress, Complete Draft.

Sutton, R.S., McAllester, D., Singh, S. & Mansour, Y. 1999. Policy gradient methods for reinforcement learning with function approximation. In S. Solla, T. Leen & K. Müller (eds.),

Advances in Neural Information Processing Systems, volume 12. MIT Press.
Available at: https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf

Ting, K.M., Washio, T., Zhu, Y. & Xu, Y. 2021. Breaking the curse of dimensionality with isolation kernel. *ArXiv*, abs/2109.14198.
Available at: https://api.semanticscholar.org/CorpusID:238215563

Walker, S. 2004. New approaches to Bayesian consistency. *The Annals of Statistics*, 32(5). ISSN 0090-5364.
Available at: http://dx.doi.org/10.1214/009053604000000409

Wang, D., Qin, X., Song, F. & Cheng, L. 2022. Stabilizing Training of Generative Adversarial Nets via Langevin Stein Variational Gradient Descent. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7):2768–2780.

Wang, D., Tang, Z., Bajaj, C. & Liu, Q. 2019. Stein Variational Gradient Descent with Matrix-Valued Kernels. *Advances in Neural Information Processing Systems*, 32:7834–7844.

Weng, L. 2018. Policy gradient algorithms. *lilianweng.github.io*.
Available at: https://lilianweng.github.io/posts/2018-04-08-policy-gradient/

Williams, R.J. 2004. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.
Available at: https://api.semanticscholar.org/CorpusID:19115634

Yan, L. & Zhou, T. 2021. Stein variational gradient descent with local approximations. *Computer Methods in Applied Mechanics and Engineering*, 386:114087.
Available at: https://doi.org/10.1016%2Fj.cma.2021.114087

Ye, M., Ren, T. & Liu, Q. 2020. Stein self-repulsive dynamics: Benefits from past samples. In *Advances in Neural Information Processing Systems*, volume 33, pages 241–252. Curran Associates, Inc.
Available at: https://proceedings.neurips.cc/paper_files/paper/2020/file/023d0a5671efd29e80b4deef8262e297-Paper.pdf

Yoon, J., Kim, T., Dia, O., Kim, S., Bengio, Y. & Ahn, S. 2018. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
Available at: https://proceedings.neurips.cc/paper_files/paper/2018/file/e1021d43911ca2c1845910d84f40aeae-Paper.pdf

Zhang, C., Butepage, J., Kjellstrom, H. & Mandt, S. 2019. Advances in Variational Inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):2008–2026.

Zhou, J. & Qiu, Y. 2023. Augmented message passing stein variational gradient descent.

Zhuo, J., Liu, C., Shi, J., Zhu, J., Chen, N. & Zhang, B. 2017. Message Passing Stein Variational Gradient Descent. In *International Conference on Machine Learning.*
Available at: https://api.semanticscholar.org/CorpusID:51877948

APPENDIX A

DERIVATIONS AND PROOFS

Proof of Lemma 2.1 (Stein's lemma).

Let $\phi(z)$ denote the probability density function of $Z \sim \mathcal{N}(0,1)$. We can use integration by parts to prove the lemma starting from the RHS as follows:

$$\begin{aligned}
\mathbb{E}\left[f'(Z)\right] &= \int_{-\infty}^{\infty} f'(z)\phi(z)dz \\
&= f(z)\phi(z)\big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} f(z)\phi'(z)dz \quad \text{(using integration by parts)} \\
&= 0 - \int_{-\infty}^{\infty} f(z)\left[-z\phi(z)\right]dz \quad (\text{since } \phi'(z) = -z\phi(z)) \\
&= \int_{-\infty}^{\infty} \phi(z)\cdot zf(z)dz \\
&= \mathbb{E}\left[Zf(Z)\right]
\end{aligned}$$

□

Proof of Corollary 2.2.

Let $Z = \frac{1}{\sigma}(X-\mu) \sim \mathcal{N}(0,1)$, and let $\tilde{f}(z) = f(\sigma z + \mu)$, where $X = \sigma Z + \mu \sim \mathcal{N}(\mu, \sigma^2)$. Note that, since we assume that f is absolutely continuous, $\tilde{f}$ is also absolutely continuous. We can now write:

$$\begin{aligned}
&\mathbb{E}\left[Z\tilde{f}(Z)\right] = \mathbb{E}\left[\tilde{f}'(z)\right] \quad \text{(by Lemma 2.1)} \\
\Longrightarrow\ &\mathbb{E}\left[\frac{1}{\sigma}(X-\mu)\tilde{f}(Z)\right] = \mathbb{E}\left[\tilde{f}'(Z)\right] \\
\Longrightarrow\ &\frac{1}{\sigma}\mathbb{E}\left[(X-\mu)f(\sigma Z+\mu)\right] = \mathbb{E}\left[f'(\sigma Z+\mu)\right] \quad (\text{by definition of } \tilde{f}) \\
\Longrightarrow\ &\frac{1}{\sigma}\mathbb{E}\left[(X-\mu)f(X)\right] = \sigma\mathbb{E}\left[f'(X)\right] \quad \text{(by the chain rule for derivatives)} \\
\Longrightarrow\ &\frac{1}{\sigma^2}\left[(X-\mu)f(X)\right] = \mathbb{E}\left[f'(X)\right]
\end{aligned}$$

□

Proof of Lemma 2.10.

$$s_p(x) = \nabla_x \log p(x) = \frac{\nabla_x p(x)}{p(x)} = \frac{\nabla_x \frac{1}{Z}\tilde{p}(x)}{\frac{1}{Z}\tilde{p}(x)} = \frac{\nabla_x \tilde{p}(x)}{\tilde{p}(x)} = s_{\tilde{p}}(x)$$

□

Proof of Corollary 2.12.

According to Definition 2.4, the Stein class $\mathcal{F}(\mathcal{T}_p)$ for a distribution P with Stein operator $\mathcal{T}_p$ is given by the class of functions $\mathcal{F}$ that satisfies:

$$\mathbb{E}_{X\sim P}\left[\mathcal{T}_p f(X)\right] = 0 \forall f \in \mathcal{F} \iff X \sim P \ .$$

If we take $\mathcal{T}_p$ to be the Langevin-Stein operator given in Equation (2.8), we may write:

$$\begin{aligned}
&\mathbb{E}_{X\sim P}\left[\mathcal{T}_p f(X)\right] = 0 \\
\iff &\mathbb{E}_{X\sim P}\left[\nabla_X \log p(X) f(X) + \nabla_X f(X)\right] = 0 \\
\iff &\int_{x\in\mathcal{X}} p(x)\left[\frac{\nabla_x p(x)}{p(x)} f(x) + \nabla_x f(x)\right] dx = 0 \\
\iff &\int_{x\in\mathcal{X}} \left[p(x)\frac{\nabla_x p(x)}{p(x)} f(x) + p(x)\nabla_x f(x)\right] dx = 0 \\
\iff &\int_{x\in\mathcal{X}} \left[f(x)\nabla_x p(x) + p(x)\nabla_x f(x)\right] dx = 0 \\
\iff &\int_{x\in\mathcal{X}} \nabla_x \left[f(x)p(x)\right] dx = 0 \quad \text{(by product rule)}
\end{aligned}$$

The final equality above is the form of the Stein class as given in Definition 2.11. □

Proof of Lemma 2.13 (Stein's identity).

$$\begin{aligned}
\mathbb{E}_{X\sim P}\left[\mathcal{T}_p f(X)\right] &= \mathbb{E}_{X\sim P}\left[\nabla_X \log p(X) f(X) + \nabla_X f(X)\right] \\
&= \int_{x\in\mathcal{X}} p(x)\left[\nabla_x \log p(x) f(x) + \nabla_x f(x)\right] dx \\
&= \int_{x\in\mathcal{X}} p(x)\left[\frac{\nabla_x p(x)}{p(x)} f(x) + \nabla_x f(x)\right] dx \\
&= \int_{x\in\mathcal{X}} \left[f(x)\nabla_x p(x) + p(x)\nabla_x f(x)\right] dx \\
&= \int_{x\in\mathcal{X}} \nabla_x \left[p(x)f(x)\right] dx \quad \text{(by the product rule for derivatives)} \\
&= 0 \quad \text{(by definition of Stein class)}
\end{aligned}$$

□

Proof of Corollary 2.14.

$$\begin{aligned}
\mathbb{E}_{X\sim Q}\left[\mathcal{T}_p f(X)\right] &= \mathbb{E}_{X\sim Q}\left[\mathcal{T}_p f(X)\right] - \mathbb{E}_{X\sim Q}\left[\mathcal{T}_q f(X)\right] \quad \text{(since } \mathbb{E}_{X\sim Q}\left[\mathcal{T}_q f(X)\right] = 0 \text{ by Stein's identity for } Q\text{)} \\
&= \mathbb{E}_{X\sim Q}\left[s_p(X) f(X)^T + \nabla_X f(X)\right] - \mathbb{E}_{X\sim Q}\left[s_q(X) f(X)^T + \nabla_X f(X)\right] \quad \text{(by Definition 2.8)} \\
&= \mathbb{E}_{X\sim Q}\left[\left\{s_p(X) - s_q(X)\right\} f(X)^T\right] \qquad \square
\end{aligned}$$

Proof of Theorem 2.15 (by contradiction).

$$\begin{aligned}
& \mathbb{E}_{X\sim Q}[\mathcal{T}_p f(X)] = 0 \\
\iff & \mathbb{E}_{X\sim Q}[(s_p(X) - s_q(X))\, f(X)] = 0 \quad \text{(by Corollary 2.14)} \\
\iff & s_p(x) - s_q(x) = 0 \; \forall x \in \mathcal{X} \quad \text{(assuming } f \neq 0 \text{ a.e)} \\
\iff & \int_{x\in\mathcal{X}} \left[\nabla_x \log p(x) - \nabla \log q(x)\right] dx = 0 \\
\iff & \int_{x\in\mathcal{X}} \nabla_x \cdot \left[\log p(x) - \log q(x)\right] dx \quad \text{(by the divergence theorem)} \\
\iff & \int_{x\in\mathcal{X}} \nabla_x \log \frac{p(x)}{q(x)} dx = 0 \\
\iff & \log \frac{p(x)}{q(x)} = 0 \\
\iff & \frac{p(x)}{q(x)} = 1 \\
\iff & p(x) = q(x)
\end{aligned}$$

This contradicts the assumption that $P \neq Q$. □

Derivation of KSD in Definition 2.17.

We can derive this form of KSD starting from the more general form of the Stein discrepancy in

Definition 2.16 as follows.

$$\begin{aligned}
\mathbb{S}(q||p) :&= \mathbb{D}_{\text{Stein}}(q, p; \mathcal{B}_{\mathcal{H}}) \\
&= \sup_{f \in \mathcal{B}_{\mathcal{H}}} \left\{ \mathbb{E}_{X \sim Q} \left[\text{trace}(\mathcal{T}_p f(X)) \right]^2 \right\} \quad \text{(by Definition 2.16)} \\
&= \sup_{f \in \mathcal{B}_{\mathcal{H}}} \langle f(\cdot), \mathbb{E}_{X \sim Q} \left[\mathcal{T}_p k(\cdot, X) \right] \rangle^2_{\mathcal{H}} \quad \text{(reproducing property)} \\
&= \| \mathbb{E}_{X \sim Q} \left[\mathcal{T}_p k(\cdot, X) \right] \|^2_{\mathcal{H}} \quad \text{(representer theorem)} \\
&= \left\langle \mathbb{E}_{X \sim Q} \left[\mathcal{T}_p k(\cdot, X) \right], \mathbb{E}_{X' \sim Q} \left[\mathcal{T}_p k(\cdot, X') \right] \right\rangle_{\mathcal{H}} \\
&= \left\langle \mathbb{E}_{X \sim Q} \left[\left(s_p(X) - s_q(X) \right) k(\cdot, X) \right], \mathbb{E}_{X' \sim Q} \left[\left(s_p(X') - s_q(X') \right) k(\cdot, X') \right] \right\rangle_{\mathcal{H}} \quad \text{(by Corollary 2.14)} \\
&= \mathbb{E}_{X, X' \sim Q} \left[\left(s_p(X) - s_q(X) \right)^T k(X, X') \left(s_p(X') - s_q(X') \right) \right] \quad \text{(reproducing property)} \qquad \square
\end{aligned}$$

Derivation of optimal posterior parameter distribution in Equation 4.6.

$$\begin{aligned}
& \nabla_q \tilde{J}(\theta) = 0 \\
\Longleftrightarrow\ & \nabla_q \left\{ \mathbb{E}_q [J(\theta) - \alpha \mathbb{D}_{\text{KL}}(q||q_0)] \right\} = 0 \\
\Longleftrightarrow\ & \nabla_q \int_\theta \left\{ q(\theta) J(\theta) - \alpha q(\theta) \left[\log q(\theta) - \log q_0(\theta) \right] \right\} d\theta = 0 \\
\Longleftrightarrow\ & \nabla_q \int_\theta \left\{ q(\theta) J(\theta) - \alpha q(\theta) \log q(\theta) + \alpha q(\theta) \log q_0(\theta) \right\} d\theta = 0 \\
\Longleftrightarrow\ & \int_\theta \left\{ \nabla_q \left[q(\theta) J(\theta) \right] - \alpha \nabla_q \left[q(\theta) \log q(\theta) \right] + \alpha \nabla_q \left[q(\theta) \log q_0(\theta) \right] \right\} d\theta = 0 \\
\Longleftrightarrow\ & \int_\theta \left\{ J(\theta) - \alpha \left[\log q(\theta) + q(\theta) \frac{\nabla_q q(\theta)}{q(\theta)} \right] + \alpha \log q_0(\theta) \right\} d\theta = 0 \\
\Longleftrightarrow\ & \int_\theta \left\{ J(\theta) - \alpha \log q(\theta) - \alpha + \alpha \log q_0(\theta) \right\} d\theta = 0 \\
\Longleftrightarrow\ & J(\theta) - \alpha \log q^*(\theta) - \alpha + \alpha \log q_0(\theta) = 0 \\
\Longleftrightarrow\ & J(\theta) = \alpha \left[\log q^*(\theta) + 1 - \log q_0(\theta) \right] \\
\Longleftrightarrow\ & \log q^*(\theta) = \frac{1}{\alpha} J(\theta) + \log q_0(\theta) - 1 \\
\Longrightarrow\ & q^*(\theta) \propto \exp \left(\frac{1}{\alpha} J(\theta) \right) q_0(\theta) \qquad \square
\end{aligned}$$

APPENDIX B

ALGORITHMS

```
Algorithm 1: Stein Variational Gradient Descent
Input: Target distribution $p(x)$, set of initial particles $\{x_i^{(0)}\}_{i=1}^n$, and a step size sequence $\{\epsilon_t\}$.
Output: A set of particles $\{x_i\}_{i=1}^n$ that approximates the target distribution.
Require: Score function of the target distribution, $s_p(x) = \nabla_x \log p(x)$ and a positive definite
  kernel $k(x, x')$
for iteration $t$ do
    for particle $i = 1$ to $n$ do
        Compute optimal update direction using Equation (2.24):
            $\hat{\phi}^*(x_i) = \frac{1}{n} \sum_{j=1}^{n} \nabla_{x_j} \log p(x_j) k(x_j, x_i) + \nabla_{x_j} k(x_j, x_i)$
        Update particle position using Equation (2.25):
            $x_i \leftarrow x_i + \epsilon_t \hat{\phi}^*(x_i)$
return Final particles $\{x_i\}_{i=1}^n$
```

Algorithm 2: Mixture-Kernel Stein Variational Policy Gradient.

```
Input: Posterior parameter distribution $q^*(\theta)$, prior distribution $q_0(\theta)$, set of initial particles
       $\{\theta_i^{(0)}\}_{i=1}^n$ and a step size sequence $\{\epsilon_t\}$.
Output: A set of particles $\{\theta_i\}_{i=1}^n$ that corresponds to a diverse set of policies $\{\pi_{\theta_i}(a|s)\}_{i=1}^n$.
Require: A set of positive definite component kernels $\{k_l(\cdot,\cdot)\}_{l=1}^m$, initial temperature $\alpha_0$ and
 a decay factor $\delta$.
for iteration $t$ do
    Compute temperature parameter $\alpha_t = \alpha_0(1-\delta)^t$.
    for particle $i = 1$ to $n$ do
        for component kernel index $l = 1$ to $m$ do
            Compute optimal update direction using Equation (4.12):
                $$\phi_l^*(\theta_i) = \frac{1}{n}\sum_{j=1}^{n} k_l(\theta_j,\theta_i)\nabla_{\theta_j}\left[\frac{1}{\alpha_t}\cdot J(\theta_j) + \log q_0(\theta_j)\right] + \nabla_{\theta_j} k_l(\theta_j,\theta_i)$$
             where $\nabla_{\theta_j} J(\theta_j)$ can be computed using any existing PG method.
        for component kernel index $l = 1$ to $m$ do
            Compute kernel weight using Equation (4.14):
                $$w_l = \frac{\|\phi_l^*(\cdot)\|}{\sum_{k=1}^m \|\phi_k^*(\cdot)\|}$$
        Compute overall optimal update direction using Equation (4.13):
                $$\phi^*(\theta_i) = \sum_{l=1}^{m} w_l \phi_l^*(\theta_i)$$
        Update particle position:
                $$\theta_i \leftarrow \theta_i + \epsilon_t \phi^*(\theta_i)$$
return Final particles $\{\theta_i\}_{i=1}^n$
```