Advancing Variational Inference: Semi-Implicit Models, Adaptive Proposals, and Functional Stein Gradients

Tobias Patrick Pielok

München, 2025

Tobias Patrick Pielok

Advancing Variational Inference: Semi-Implicit Models, Adaptive Proposals, and Functional Stein Gradients

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

Eingereicht am 12.12.2025

Erster Berichterstatter: Prof. Dr. Bernd Bischl
Zweiter Berichterstatter: Prof. Dr. David Rügamer
Dritter Berichterstatter: Prof. Dr. Florian Büttner

Tag der Disputation: 13.03.2026

Acknowledgments

I would like to express my sincere gratitude to ...

- *... Prof. Dr. Bernd Bischl, for his support, supervision, and for granting me academic freedom.*
- *... Prof. Dr. David Rügamer, for his willingness to dedicate his valuable time as the second reviewer of this thesis, but mostly for always being available to discuss my ideas and for his unwavering support throughout my research.*
- *... Prof. Dr. Florian Büttner, for his willingness to dedicate his valuable time as the third reviewer of this thesis.*
- *... PD Dr. Fabian Scheipl, for his availability to be part of my doctoral committee, but mostly, for his inspiring example as a mentor.*
- *... Prof. Dr. Thomas Nagler, for his availability to be part of my doctoral committee.*
- *... My parents, for always believing in me.*

Summary

Variational inference (VI) is a widely used framework for approximate Bayesian inference, offering computational scalability but often relying on overly simplistic approximations. These limitations reduce the ability of standard VI methods to capture complex posterior distributions. This thesis addresses these challenges by developing methods that improve the expressiveness and stability of VI through function-space optimization, importance sampling, and kernel-based gradient estimation.

The first part introduces a functional formulation of VI, Stein Functional Variational Gradient Descent, which directly optimizes distributions over functions using gradients derived from Stein's identity. This approach enables accurate predictive inference in overparameterized models, such as Bayesian neural networks, where the posterior distribution itself is not very informative due to the large number of parameters and overparameterization.

In contrast, the second part of the thesis focuses on improving semi-implicit variational inference (SIVI), where the goal shifts from predictive inference to closely approximating complex posterior densities. A new unbiased training objective for SIVI is introduced, which replaces MCMC-based inner loops with importance sampling from learned proposal distributions. While this yields a tractable and fully differentiable estimator, further improvements are achieved by incorporating kernelized path gradients and a bias-correcting importance sampling correction. The resulting method combines the strengths of proposal learning and nonparametric smoothing, improving both the stability and accuracy of posterior approximation.

Together, these contributions offer a unified perspective on advancing variational inference beyond conventional approximations. By integrating functional inference, adaptive proposal mechanisms, and kernel-based estimators, the proposed methods enhance the fidelity and practical utility of Bayesian inference in modern machine learning settings.

Zusammenfassung

Die Variationsinferenz (VI) ist ein etabliertes Verfahren der approximativen Bayes'schen Inferenz. Ihr Hauptvorteil liegt in der hohen rechnerischen Skalierbarkeit, doch wird die Genauigkeit von VI-Methoden häufig durch zu restriktive Approximationsannahmen gemindert. Insbesondere werden durch diese Annahmen häufig die Fähigkeit herkömmlicher VI-Methoden, komplexe Posteriorverteilungen adäquat zu erfassen, eingeschränkt. Diese Herausforderungen werden in der vorliegenden Dissertation adressiert, indem Verfahren entwickelt werden, die die Repräsentationsfähigkeit und Stabilität der Variationsinferenz durch Optimierung im Funktionsraum, gewichtete Stichproben zur Varianzreduktion und kernelbasierte Gradientenabschätzung verbessern.

Im ersten Teil der Arbeit wird eine funktionale Formulierung der Variationsinferenz eingeführt, Stein Functional Variational Gradient Descent, bei der Verteilungen über Funktionen direkt mithilfe von Gradienten optimiert werden, die aus der Stein'schen Identität hergeleitet sind. Durch diesen Ansatz wird eine präzise prädiktive Inferenz in überparametrisierten Modellen, insbesondere Bayes'schen neuronalen Netzen, ermöglicht, bei denen die Posteriorverteilung aufgrund der großen Anzahl an Parametern und der Überparametrisierung nur begrenzt informativ ist.

Im Gegensatz dazu werden im zweiten Teil der Dissertation neue Methoden für die sogenannte Semi-Implicit Variational Inference (SIVI) entwickelt, wobei der Schwerpunkt von prädiktiver Inferenz hin zu einer genaueren Approximation komplexer posteriorer Dichten verlagert wird. Es wird ein neues unverzerrtes Verfahren eingeführt, das darauf beruht, einen MCMC-basierten Berechnungsschritt durch gewichtete Stichprobenverfahren mit gelernten Vorschlagsverteilungen zu ersetzen. Dadurch entsteht ein effizient berechenbarer und vollständig differenzierbarer Gradientenschätzer. Weitere Verbesserungen werden durch die Einbindung von kernelbasierten Richtungsableitungen entlang reparametrisierter Pfade sowie einer Verzerrungskorrektur mit Hilfe von gewichteten Stichproben erzielt. Hierdurch werden die Vorteile adaptiver Vorschlagsverteilungen und nichtparametrischer Glättung in der resultierenden Methode vereint, wodurch sowohl die Stabilität als auch die Genauigkeit der posterioren Approximation erhöht werden.

Durch diese methodischen Innovationen entsteht eine einheitliche Perspektive auf die Weiterentwicklung der Variationsinferenz über konventionelle Approximationen hinaus. Die vorgeschlagenen Verfahren, die funktionale Inferenz, adaptive Vorschlagsmechanismen und kernelbasierte Schätzer integrieren, verbessern sowohl die Genauigkeit als auch die praktische Anwendbarkeit der Bayes'schen Inferenz in modernen maschinellen Lernumgebungen.

Contents

List of Figures

List of Tables

Notation

Throughout this thesis, the following conventions will be used:

General Conventions. Scalars are denoted by lowercase letters (e.g., x), vectors by bold lowercase letters (e.g., $\boldsymbol{x}$, $\boldsymbol{\theta}$), and matrices by bold uppercase letters (e.g., $\boldsymbol{X}$). Random variables are typically denoted in bold (e.g., $\boldsymbol{z}$, $\boldsymbol{\epsilon}$). Subscripts indicate elements of a vector (e.g., x_i is the i-th element of vector $\boldsymbol{x}$) or matrix entries (e.g., X_{ij}). Superscripts in parentheses enumerate distinct instances (e.g., $\boldsymbol{z}^{(i)}$ is the i-th sample). Superscripts in square brackets denote iteration indices (e.g., $\boldsymbol{\theta}^{[t]}$ is the parameter vector at iteration t). A tilde denotes a realized random variable (e.g., $\widetilde{\boldsymbol{u}}$ is a realization of $\boldsymbol{u}$).

Probability and Distributions.

P, Q	Probability measures (distributions)
p, q	Probability density functions
π	Base distribution (e.g., in normalizing flows)
$\boldsymbol{z}$	Random variable
$\boldsymbol{x}, \boldsymbol{y}$	Input and output random variables (function spaces)
$\boldsymbol{\epsilon}$	Noise random variable
$\mathbb{E}[\cdot]$	Expectation operator
$\mathbb{E}_{\boldsymbol{z}\sim Q}[\cdot]$	Expectation with respect to distribution Q
$\mathbb{E}_{\boldsymbol{z}\sim q}[\cdot]$	Expectation with respect to density q
$\sim$	Distributed according to (e.g., $\boldsymbol{z} \sim Q$)
$\mid$	Condition bar (used in conditionals, e.g., $q(\boldsymbol{z} \mid \boldsymbol{\epsilon})$)

Bayesian Inference.

$\boldsymbol{z}$	Latent random variable
$\boldsymbol{u}$	Observable random variable
$\widetilde{\boldsymbol{u}}$	Realization of observable $\boldsymbol{u}$
Π	Prior distribution (measure)
$p(\boldsymbol{z})$	Prior density
$P(\cdot \mid \boldsymbol{z})$	Observation model (probability kernel)

$L(\widetilde{\boldsymbol{u}} \mid \boldsymbol{z})$	Likelihood function
$l(\widetilde{\boldsymbol{y}} \mid \widetilde{\boldsymbol{x}}, \boldsymbol{z})$	Individual likelihood (supervised setting)
Λ	Joint distribution of $(\boldsymbol{u}, \boldsymbol{z})$
$\Lambda_{\mathcal{U}}$	Prior predictive distribution (marginal of $\boldsymbol{u}$)
$Z(\widetilde{\boldsymbol{u}})$	Evidence (marginal likelihood)
$\Pi(\cdot \mid \widetilde{\boldsymbol{u}})$	Posterior distribution given observation $\widetilde{\boldsymbol{u}}$
$p(\boldsymbol{z} \mid \widetilde{\boldsymbol{u}})$	Posterior density
$\Lambda_{\mathcal{U}}^{\text{post}}$	Posterior predictive distribution
$P_{\boldsymbol{xy}}$	Joint data-generating distribution (supervised setting)
$P_{\boldsymbol{x}}$	Marginal distribution over inputs

Variational Inference.

$\mathcal{I}_{\boldsymbol{\lambda}}$	VI algorithm with hyperparameters $\boldsymbol{\lambda}$
$\boldsymbol{\lambda}$	Hyperparameters of VI algorithm
$\boldsymbol{\theta}$	Variational parameters
$Q_{\boldsymbol{\theta}}$	Parameterized variational distribution
$q_{\boldsymbol{\theta}}$	Variational density function
$\mathcal{Q}$	Variational family
$\mathcal{L}$	Loss function or divergence
$\nabla_{\boldsymbol{\theta}}$	Gradient with respect to $\boldsymbol{\theta}$
$\boldsymbol{d}, \widehat{\boldsymbol{d}}$	True and estimated gradient
η	Learning rate

Divergences and Distances.

$\mathrm{D}_{\mathrm{KL}}(Q \,\|\, P)$	Kullback–Leibler divergence from Q to P
$\mathrm{D}_{f}(Q \,\|\, P)$	f-divergence
$\mathrm{D}_{\mathrm{Fisher}}(Q \,\|\, P)$	Fisher divergence
$\mathrm{D}_{\mathrm{Stein},\mathcal{F}}(Q \,\|\, P)$	Stein discrepancy
$\mathrm{D}_{\mathrm{KF}}(Q \,\|\, P)$	Kernelized Fisher divergence
$\mathrm{D}_{\mathrm{KSD}}(Q \,\|\, P)$	Kernelized Stein discrepancy (KSD)
$\mathrm{D}_{\mathrm{MMD}}(Q, P)$	Maximum mean discrepancy (MMD)
$\mathrm{D}_{W_p}(Q, P)$	Wasserstein distance of order p

Transformations and Functions.

$f_{\boldsymbol{\theta}}, h_{\boldsymbol{\theta}}$	Parametric functions (e.g., neural networks)
$T, T_{\boldsymbol{\theta}}$	Transformation or generator function
f, g, h	Generic functions
$\boldsymbol{s}_p(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z})$	Score function of density p
$k(\cdot, \cdot)$	Kernel function
$\mathcal{H}_k$	Reproducing kernel Hilbert space (RKHS)
$\langle \cdot, \cdot \rangle_{\mathcal{H}}$	Inner product in Hilbert space $\mathcal{H}$
$\| \cdot \|_{\mathcal{H}}$	Norm in Hilbert space $\mathcal{H}$

Sets and Spaces.

$\mathbb{R}$	Real numbers
$\mathbb{R}^d$	d-dimensional Euclidean space
$\dim(\boldsymbol{x})$	Dimension (number of components) of vector $\boldsymbol{x}$
$\mathbb{N}$	Natural numbers
$\mathcal{Z}$	Latent space
$\mathcal{U}$	Space of observables
$\mathcal{X}, \mathcal{Y}$	Input and output spaces (for functions)
Θ	Parameter space
Ω	Sample space in probability theory
$\mathcal{B}$	σ-algebra
λ	Lebesgue measure

Data and Supervised Learning.

$\mathcal{D}_{\text{train}}$	Training data set
$\mathcal{D}_{\text{test}}$	Test data set
N	Number of training observations
M	Number of test observations
$\widehat{h}$	Probabilistic model

Abbreviations

Throughout this thesis, the following abbreviations are used:

CLT	Central limit theorem
CNF	Conditional normalizing flow
ELBO	Evidence lower bound
ESS	Effective sample size
FVI	Functional variational inference
GP	Gaussian process
IS	Importance sampling
KL	Kullback–Leibler (divergence)
KSD	Kernel Stein discrepancy
KSIVI	Kernel semi-implicit variational inference
LLN	Law of large numbers
MC	Monte Carlo
MCMC	Markov chain Monte Carlo
MMD	Maximum mean discrepancy
NLL	Negative log likelihood
RKHS	Reproducing kernel Hilbert space
SIVI	Semi-implicit variational inference
SVGD	Stein variational gradient descent
VI	Variational inference

Part I.

Introduction and Background

1. Introduction

1.1. Motivation and Scope

At its core, this thesis is concerned with a central question in probabilistic inference:

How can we efficiently perform inference with complex, high- or even infinite-dimensional continuous probability distributions?

Continuous probability distributions play a central role in expressing uncertainty and modeling subjective beliefs in Bayesian inference (Gelman et al., 2013; O'Hagan and Forster, 2004), and in describing physical systems via the Boltzmann distribution in statistical mechanics (Landau and Lifshitz, 1980; Pathria and Beale, 2022). Their practical relevance is based on the fact that many models in machine learning (ML) and the natural sciences naturally represent uncertainty in continuous spaces (Gelman et al., 2013; Walck, 1996). Working with such models typically requires computing expectations to derive predictions, quantify risks, and support policy choices (Berger, 1985; Robert and Casella, 2004). Yet these expectations are rarely available in closed form, especially in high-dimensional or otherwise complex settings. We therefore rely on Monte Carlo methods and related sampling schemes (Robert and Casella, 2004; Owen, 2013). While asymptotically accurate, basic sampling approaches can become inefficient or even intractable as dimensionality increases (Neal, 2011). In contrast, variational inference (VI) offers a scalable alternative by treating inference as an optimization problem (Jordan et al., 1998; Blei et al., 2017). However, traditional VI often uses approximating families that are too restrictive to capture the structure of complex Bayesian posteriors (Zhang et al., 2019).

This thesis develops methods that advance VI for continuous distributions along two complementary directions. The first direction examines the question:

In overparameterized models where individual parameters are not meaningful, can we perform variational inference directly over functions defined on continuous spaces?

A canonical example is Bayesian neural networks, whose posterior distributions over the parameters are difficult to interpret due to overparameterization (MacKay, 1992; Neal, 1996; Izmailov et al., 2021). In this setting, the distributions over predictions induced by the posterior distributions are typically of primary interest. We introduce a functional approach to VI that optimizes distributions over functions, using gradients derived from Stein's identity. This enables us to train a function generator approximating the posterior distribution over functions without learning a posterior distribution in the overparameterized parameter space.

The second direction focuses on inference in high-dimensional continuous distributions on $\mathbb{R}^d$ with differentiable densities. Here the central challenge is:

How can we perform efficient VI for complex absolutely continuous probability distributions given by differentiable density functions without restraining ourselves to simple variational families?

Semi-implicit variational inference (SIVI; Yin and Zhou 2018) offers a highly expressive approximation family via its hierarchical constructions. However, many existing SIVI techniques can be computationally demanding or unstable in practice. This thesis proposes new strategies that improve the efficiency and robustness of SIVI for high-dimensional problems, reducing computational costs while maintaining or improving accuracy. These advances provide flexible and reliable inference tools for complex continuous distributions.

While the broader literature also includes hybrid methods that blend VI with MCMC or score-based diffusion techniques (e.g., Midgley et al., 2023; Vargas et al., 2024; He et al., 2025), these approaches are not the focus of this thesis. They typically rely on iterative refinement and are often designed for settings where the target distribution contains high-probability regions separated by low-density barriers, whereas we focus on computationally efficient one-shot generators for targets which do not exhibit such severe separations.

Outline. The structure of this thesis is as follows:

- **Part I: Introduction and Background** (Chapters 1–2) covers motivation and core theoretical tools, including probability theory, reproducing kernel Hilbert space (RKHS) theory, divergences, and a survey of variational inference approaches.
- **Part II: Functional Variational Inference** (Chapter 3) develops a kernel-based functional VI method for predictive inference, allowing efficient optimization in high-dimensional models via analytic solutions in RKHS.
- **Part III: Semi-Implicit Variational Inference** (Chapters 4–5) presents advances in SIVI. First, a new unbiased objective replaces an inner Markov chain Monte Carlo (MCMC) training step with importance sampling via adaptive proposal distributions modeled by conditional normalizing flows, improving efficiency. Second, a kernelized path gradient method further reduces computational cost and improves accuracy.
- **Part IV: Conclusion** (Chapter 6) summarizes main insights, implications for VI, and future research directions.

2. Methodological and General Background

2.1. Probability and Integration

This chapter summarizes the most important definitions and establishes the notation used throughout the thesis. We adopt a general, measure-theoretic formulation of probability, which provides a flexible framework for defining and manipulating distributions over both finite-dimensional spaces (e.g., $\mathbb{R}^d$) and infinite-dimensional spaces (e.g., function spaces). This allows us to treat classical variational inference and its functional extensions from a unified mathematical perspective.

Probability Spaces and Random Variables. A *probability space* is a triple $(\Omega, \mathcal{B}_\Omega, \mathbb{P})$, where

- Ω is the sample space,
- $\mathcal{B}_\Omega$ is a σ-algebra of measurable subsets of Ω,
- $\mathbb{P}$ is a probability measure with $\mathbb{P}(\Omega) = 1$.

A *random variable* is a measurable function

$$\boldsymbol{z} : \Omega \to \mathcal{Z} \tag{2.1}$$

from the probability space $(\Omega, \mathcal{B}_\Omega, \mathbb{P})$ into a measurable space $(\mathcal{Z}, \mathcal{B}_\mathcal{Z})$, where $\mathcal{B}_\mathcal{Z}$ is a σ-algebra on $\mathcal{Z}$. In our setting, the so-called *state space* $\mathcal{Z}$ is either a subset of $\mathbb{R}^d$ with $d \in \mathbb{N}$, or a function space. The *distribution* of $\boldsymbol{z}$ is the *pushforward measure* of $\mathbb{P}$ under $\boldsymbol{z}$, denoted $Q = \boldsymbol{z}_\# \mathbb{P}$. It is defined by

$$Q(A) = \mathbb{P}(\boldsymbol{z}^{-1}(A)) \quad \text{for all } A \in \mathcal{B}_\mathcal{Z}, \tag{2.2}$$

where $\boldsymbol{z}^{-1}(A) := \{\omega \in \Omega : \boldsymbol{z}(\omega) \in A\}$. We will write $\boldsymbol{z} \sim Q$ to indicate that the distribution of $\boldsymbol{z}$ is Q. When $\mathcal{Z}$ is a function space mapping from an index set $\mathcal{X} \subseteq \mathbb{R}^p$ with $p \in \mathbb{N}$ to a measurable space $(\mathcal{Y}, \mathcal{B}_\mathcal{Y})$, with $\mathcal{Y} \subseteq \mathbb{R}^m$, $m \in \mathbb{N}$, we equip $\mathcal{Z}$ with a σ-algebra $\mathcal{B}_\mathcal{Z}$ chosen so that all evaluation maps

$$\mathrm{ev}_{\boldsymbol{x}} : \mathcal{Z} \to \mathcal{Y}, \qquad \mathrm{ev}_{\boldsymbol{x}}(f) = f(\boldsymbol{x}), \tag{2.3}$$

are measurable. In this case, a random variable $\boldsymbol{z} : \Omega \to \mathcal{Z}$ is called a *stochastic process*, and for each $\boldsymbol{x} \in \mathcal{X}$, the map $\boldsymbol{z}_{\boldsymbol{x}} := \mathrm{ev}_{\boldsymbol{x}} \circ \boldsymbol{z}$ is a random variable with state space $\mathcal{Y}$.

In practice, one often constructs $\boldsymbol{z}$ by transforming a simpler random variable. If $\boldsymbol{\epsilon}$ is a random variable with known distribution π and $T : \mathcal{E} \to \mathcal{Z}$ is measurable, then $\boldsymbol{z} = T(\boldsymbol{\epsilon})$ has distribution $Q = T_\# \pi = (T \circ \boldsymbol{\epsilon})_\# \mathbb{P}$. This viewpoint is central in many VI methods, where π is called the base distribution (for example, a standard Gaussian) and Q is implicitly defined via $\boldsymbol{z} = T(\boldsymbol{\epsilon})$.

A probability measure Q is said to be *absolutely continuous* with respect to a reference measure μ if $\mu(A) = 0$ implies $Q(A) = 0$ for all measurable A. In this case, the Radon–Nikodym theorem (Billingsley, 1995) guarantees the existence of a μ-almost everywhere unique, measurable, nonnegative function called the *Radon–Nikodym derivative*, $\frac{dQ}{d\mu}$, such that

$$Q(A) = \int_A \frac{dQ}{d\mu}(\boldsymbol{z})\, \mu(d\boldsymbol{z}), \qquad A \in \mathcal{B}_{\mathcal{Z}}. \tag{2.4}$$

When the reference measure μ is the Lebesgue measure λ (which we assume by default when none is specified), we call the Radon–Nikodym derivative the *probability density function*[1] q of Q, and we will, to simplify notation, sometimes (slightly abusively) write $\boldsymbol{z} \sim q$ instead of $\boldsymbol{z} \sim Q$. However, this general formulation also enables the use of non-standard reference measures. For example, in the case of stochastic processes, a density with respect to Lebesgue measure does not exist, since there is no natural Lebesgue measure in infinite dimensions (Bogachev, 1998); nevertheless, it is often possible to define a Radon–Nikodym derivative with respect to the distribution of another stochastic process (Schervish, 1995), which is often assumed in the context of functional variational inference.

Expectation and Integration. As outlined in the introduction (Section 1.1), expectations are essential for both inference and decision making. In variational inference, they are particularly important because variational objectives are usually expressed in terms of expectations.

In general, for any measurable function $f : \mathcal{Z} \to \mathbb{R}$ and σ-finite measure μ on $(\mathcal{Z}, \mathcal{B}_{\mathcal{Z}})$, we write

$$\mathbb{E}_{\mu}[f] = \int_{\mathcal{Z}} f(\boldsymbol{z})\, \mu(d\boldsymbol{z}). \tag{2.5}$$

When $\mu = Q$ is a probability measure, we write $\mathbb{E}_{\boldsymbol{z}\sim Q}[f(\boldsymbol{z})] = \mathbb{E}_{Q}[f]$. This is the expectation of $f(\boldsymbol{z})$ under the probability measure Q. If Q admits a probability density function q, then

$$\mathbb{E}_{\boldsymbol{z}\sim Q}[f(\boldsymbol{z})] = \mathbb{E}_{\boldsymbol{z}\sim q}[f(\boldsymbol{z})] = \int_{\mathcal{Z}} f(\boldsymbol{z})\, q(\boldsymbol{z})\, \lambda(d\boldsymbol{z}) \text{ with Lebesgue measure } \lambda. \tag{2.6}$$

In practice, expectations such as $\mathbb{E}_{\boldsymbol{z}\sim Q}[f(\boldsymbol{z})]$ are often approximated using *Monte Carlo integration*. By the law of large numbers (LLN), under the assumption $\mathbb{E}_{\boldsymbol{z}\sim Q}[|f(\boldsymbol{z})|] < \infty$, the empirical average of independent and identically distributed samples $\boldsymbol{z}^{(i)} \overset{\text{i.i.d.}}{\sim} Q$ with $i \in \mathbb{N}$ converges almost surely to the true expectation:

$$\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{z}^{(i)}) \xrightarrow{\text{a.s.}} \mathbb{E}_{\boldsymbol{z}\sim Q}[f(\boldsymbol{z})] \quad \text{as } n \to \infty, \tag{2.7}$$

i.e., the set of infinite sequences $\left(\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots\right)$ for which the empirical average does not converge to $\mathbb{E}_{\boldsymbol{z}\sim Q}[f(\boldsymbol{z})]$ has probability zero. Given i.i.d. samples $\{\boldsymbol{z}^{(i)}\}_{i=1}^{n}$ drawn from Q, we therefore can estimate the expectation by the sample average which we call the *Monte Carlo* (MC) estimator:

$$\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{z}^{(i)}) \approx \mathbb{E}_{\boldsymbol{z}\sim Q}[f(\boldsymbol{z})]. \tag{2.8}$$

[1] As commonly done in the literature, we use the terms *density* and *probability density function* interchangeably.

Since in our setting the samples are i.i.d. and the expectation is a linear operator, it directly follows that the Monte Carlo estimator is unbiased, and by the LLN it is also a consistent estimator. For $\mathcal{Z} \subset \mathbb{R}^d$ with $d \in \mathbb{N}$ the convergence rate of the Monte Carlo estimator is $\mathcal{O}(1/\sqrt{n})$ which follows from the Central Limit Theorem (CLT), under the assumption that the variance of the estimator is finite, i.e., $\mathrm{Var}_{\boldsymbol{z}\sim Q}[f(\boldsymbol{z})] = \mathbb{E}_{\boldsymbol{z}\sim Q}[(f(\boldsymbol{z}) - \mathbb{E}_{\boldsymbol{z}\sim Q}[f(\boldsymbol{z})])^2] < \infty$. In contrast, the central limit theorems for stochastic processes do not guarantee a universal convergence rate of $\mathcal{O}(1/\sqrt{n})$. Convergence may be arbitrarily slow, and nontrivial rates only arise under strong regularity assumptions (Araujo and Giné, 1980). However, even in the finite-dimensional case, Monte Carlo estimates can exhibit high variance, particularly in high-dimensional settings or when $f(\boldsymbol{z})$ has heavy tails or substantial variability with respect to Q (Robert and Casella, 2004; Owen, 2013). This in turn increases the constants hidden in the $\mathcal{O}(1/\sqrt{n})$ scaling, which can severely limit the practical feasibility of Monte Carlo methods despite their asymptotic convergence properties.

Importance Sampling. Classical variance-reduction techniques for Monte Carlo methods include control variates, antithetic sampling, stratified sampling, and quasi-Monte Carlo methods (Owen, 2013). Among these, when the target distribution P admits a density p, *importance sampling* (IS) stands out for its simplicity and effectiveness and also guides the design and analysis of variational inference methods. Here, we introduce a proposal distribution Q with density q from which sampling and density evaluation are efficient. Crucially, we require that $q(\boldsymbol{z}) = 0$ implies $p(\boldsymbol{z}) = 0$ for λ-almost every $\boldsymbol{z} \in \mathcal{Z}$. Under this assumption, we can rewrite[2] the expectation as

$$\mathop{\mathbb{E}}_{\boldsymbol{z}\sim p}[f(\boldsymbol{z})] = \int_{\mathrm{supp}(Q)} f(\boldsymbol{z}) \frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}\, q(\boldsymbol{z})\, \lambda(d\boldsymbol{z}) = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim q}\left[f(\boldsymbol{z}) \cdot \frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}\right] \approx \frac{1}{n}\sum_{i=1}^{n} f\left(\boldsymbol{z}^{(i)}\right) \frac{p\left(\boldsymbol{z}^{(i)}\right)}{q\left(\boldsymbol{z}^{(i)}\right)}, \tag{2.9}$$

where $\boldsymbol{z}^{(i)} \overset{\text{i.i.d.}}{\sim} Q$ and $\mathrm{supp}(Q)$ is the support of Q, i.e., the set of all $\boldsymbol{z} \in \mathcal{Z}$ for which $q(\boldsymbol{z}) > 0$. By construction, the IS estimator inherits the unbiasedness and consistency properties of the Monte Carlo estimator. We call the quantity $\frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}$ the *importance weight* of $\boldsymbol{z}$ with respect to Q; it corrects for the mismatch between the target and proposal distributions.

Formally, the variance-minimizing (unnormalized) proposal (Owen, 2013) is

$$q^*(\boldsymbol{z}) = p(\boldsymbol{z})\, |f(\boldsymbol{z})|. \tag{2.10}$$

In general, q^* cannot be used directly: its normalizing constant is unknown, or it may not even be possible to normalize it into a proper density. Instead, we choose a tractable proposal q whose shape mimics q^*, in the sense that q assigns high mass to regions where q^* is large and does not have lighter tails. If q fails to assign enough probability mass to those regions, the resulting importance weights become highly variable. This can lead to estimators with large or even infinite variance, which in turn can result in cases where performance is worse than that of simple Monte Carlo estimation.

We also note that importance sampling applies in situations where sampling from P is infeasible but its density is evaluable. In this case, the importance weights can provide insights into the quality of the proposal distribution Q with respect to P, leading to the following indicator.

[2] This identity is the probabilistic analogue of the change-of-variables formula for integrals.

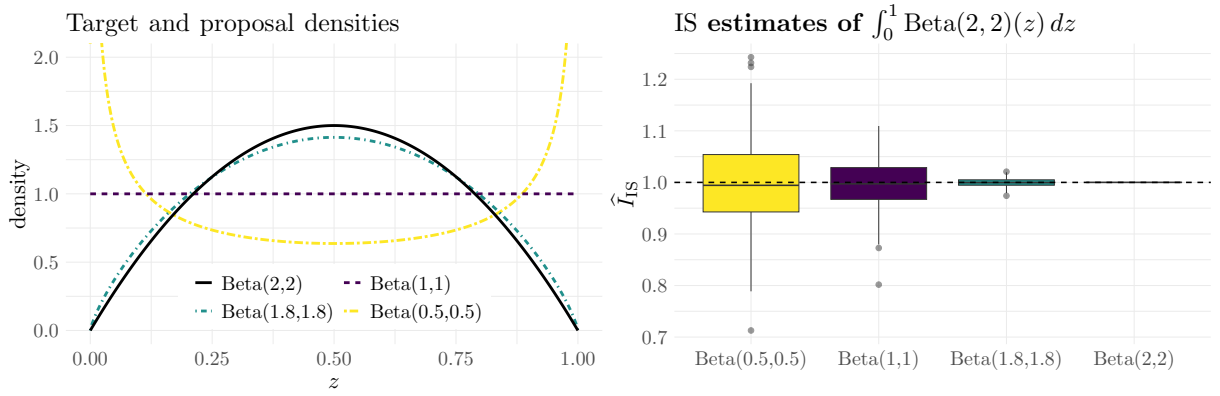

Figure 2.1.: Illustration of importance sampling (IS): left, target and proposal Beta densities; right, distribution of IS estimates across proposals, with each estimate based on $n = 100$ samples and $R = 500$ repeated trials per proposal.

Effective Sample Size. The *effective sample size* (ESS; Liu, 2001) is a widely used diagnostic that quantifies how many equally weighted samples would provide an estimator of comparable quality to a given set of importance-weighted samples.

Given importance weights $w^{(i)} = p(\boldsymbol{z}^{(i)})/q(\boldsymbol{z}^{(i)})$ and normalized weights

$$\overline{w}^{(i)} = \frac{w^{(i)}}{\sum_{j=1}^{n} w^{(j)}}, \qquad \boldsymbol{z}^{(i)} \overset{\text{i.i.d.}}{\sim} Q, \tag{2.11}$$

the ESS is typically estimated by

$$\text{ESS} = \frac{1}{\sum_{i=1}^{n} (\overline{w}^{(i)})^2} \in [1, n]. \tag{2.12}$$

The *relative* ESS $:= \text{ESS}/n \in [1/n, 1]$ is often used for comparison between different approximations. It attains its maximum value 1 when all weights are equal (i.e., when sampling directly from the target distribution P), and becomes $1/n$ as one normalized weight becomes one and the others equal zero.

Note that ESS does not account for the integrand f and therefore does not reflect the actual variance of the IS estimator $1/n \cdot \sum_{i=1}^{n} f(\boldsymbol{z}^{(i)}) w(\boldsymbol{z}^{(i)})$. Hence, ESS is a useful but limited indicator of how well the proposal matches the target distribution, not of the variance of a particular importance sampling estimate.

Importance Sampling Illustration. In the setting illustrated in Figure 2.1, the goal is to estimate the expectation $\mathbb{E}_{\boldsymbol{z}\sim p}[f(\boldsymbol{z})]$, where $p = \mathcal{U}([0,1])$, i.e., the uniform distribution on $[0,1]$ and f is the density of the Beta$(2,2)$ distribution. Since the expectation with respect to $\mathcal{U}([0,1])$ is equal to the Lebesgue integral of f over $[0,1]$ and the support of the beta distribution is $(0,1)$, it directly follows that the expectation is equal to 1.

To perform importance sampling, we introduce a proposal density q whose support covers $(0,1)$ and express the expectation as $\mathbb{E}_{\boldsymbol{z}\sim q}[f(\boldsymbol{z}) \cdot 1/q(\boldsymbol{z})]$. In Figure 2.1, the left panel displays the target

density along with several choices of proposal q, and the right panel shows the distribution of importance sampling estimates obtained from these proposals.

If the proposal q does not resemble the optimal choice q^*, which in this specific case is f (since $f \geq 0$ is a density and p is constant 1), the resulting importance weights $1/q(\boldsymbol{z})$ can be highly variable. This results in high-variance or even unreliable estimates. As q becomes closer to the optimal q^*, the variance of the importance sampling estimator decreases significantly. In fact, if q is chosen to be exactly equal to f, the estimator achieves zero variance, meaning every sample gives the correct value. This zero-variance proposal exists in this example because f is nonnegative and q^* has finite integral over the support of p.

This example highlights both the consequences of using poor proposals and the substantial gain in reliability and accuracy as q approaches the optimal weight function. Also note, as discussed in Section 2.1, ESS/n is only 1 when Q is the uniform target distribution, which however is not the variance-minimizing proposal here.

Vector-Valued Functions. The ideas introduced in this integration chapter extend naturally to vector-valued functions $\boldsymbol{f} : \mathcal{Z} \to \mathbb{R}^m$. Here, expectations are defined componentwise, i.e.,

$$\mathbb{E}_{\boldsymbol{z}\sim Q}[\boldsymbol{f}(\boldsymbol{z})] = \left(\mathbb{E}_{\boldsymbol{z}\sim Q}[f_1(\boldsymbol{z})], \ldots, \mathbb{E}_{\boldsymbol{z}\sim Q}[f_m(\boldsymbol{z})]\right)^\top. \tag{2.13}$$

The Monte Carlo estimator is then defined componentwise as

$$\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{f}(\boldsymbol{z}^{(i)}) = \left(\frac{1}{n}\sum_{i=1}^{n} f_1(\boldsymbol{z}^{(i)}), \ldots, \frac{1}{n}\sum_{i=1}^{n} f_m(\boldsymbol{z}^{(i)})\right)^\top \approx \mathbb{E}_{\boldsymbol{z}\sim Q}[\boldsymbol{f}(\boldsymbol{z})] \text{ with } \boldsymbol{z}^{(i)} \overset{\text{i.i.d.}}{\sim} Q. \tag{2.14}$$

The Monte Carlo estimator is unbiased by construction and consistent by the LLN which holds if $\mathbb{E}_{\boldsymbol{z}\sim Q}\left[\|\boldsymbol{f}(\boldsymbol{z})\|_2\right] < \infty$. For $\mathcal{Z} \subset \mathbb{R}^d$ with $d \in \mathbb{N}$ the convergence rate of the MC estimator is analogously $\mathcal{O}(1/\sqrt{n})$ by the multivariate CLT which holds if the variance of each component of $\boldsymbol{f}$ is finite, i.e., $\mathbb{E}_{\boldsymbol{z}\sim Q}\left[\|\boldsymbol{f}(\boldsymbol{z})\|_2^2\right] < \infty$. However for importance sampling, proposal design becomes more involved, as there is generally no single distribution that simultaneously minimizes the estimator variance for all components of $\boldsymbol{f}$ (Owen, 2013).

2.2. Hilbert Spaces

Throughout this thesis, we assume that expectations of gradients (which can be treated as vector-valued functions) are well-defined. This section establishes the theoretical foundations necessary for this assumption.

Hilbert Spaces. A *Hilbert space* $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ is a complete inner product space, where the inner product induces a norm for all $\boldsymbol{f} \in \mathcal{H}$ by

$$\|\boldsymbol{f}\|_{\mathcal{H}} = \sqrt{\langle \boldsymbol{f}, \boldsymbol{f}\rangle_{\mathcal{H}}}. \tag{2.15}$$

Completeness means that every Cauchy sequence in $\mathcal{H}$ converges to an element of $\mathcal{H}$. Hilbert spaces generalize Euclidean spaces to possibly infinite-dimensional settings while retaining geometric notions such as orthogonality, projections, and expansions in orthonormal bases.

The Space $L^2(\mu;\mathbb{R}^m)$. For a measure μ on a measurable space $(\mathcal{Z},\mathcal{B})$, the space

$$L^2(\mu;\mathbb{R}^m) = \left\{ \boldsymbol{f} : \mathcal{Z} \to \mathbb{R}^m \text{ measurable} \;\middle|\; \int_{\mathcal{Z}} \|\boldsymbol{f}(\boldsymbol{z})\|_2^2 \, d\mu(\boldsymbol{z}) < \infty \right\} \tag{2.16}$$

is a Hilbert space when equipped with the inner product

$$\langle \boldsymbol{f}, \boldsymbol{g} \rangle_{L^2(\mu)} = \int_{\mathcal{Z}} \boldsymbol{f}(\boldsymbol{z})^\top \boldsymbol{g}(\boldsymbol{z}) \, d\mu(\boldsymbol{z}), \tag{2.17}$$

defined for all $\boldsymbol{f},\boldsymbol{g} \in L^2(\mu;\mathbb{R}^m)$, and the corresponding norm $\|\boldsymbol{f}\|_{L^2(\mu;\mathbb{R}^m)} = \sqrt{\langle \boldsymbol{f},\boldsymbol{f}\rangle_{L^2(\mu;\mathbb{R}^m)}}$. Functions that differ only on a set A with $\mu(A)=0$ are identified as the same element, ensuring that $\langle \boldsymbol{f},\boldsymbol{f}\rangle_{L^2(\mu;\mathbb{R}^m)} = 0$ implies $\boldsymbol{f} = \boldsymbol{0}$ μ-almost everywhere. Since every probability measure Q is a measure, each Q induces its own Hilbert space $L^2(Q;\mathbb{R}^m)$.

Reproducing Kernel Hilbert Spaces. A *reproducing kernel Hilbert space* (RKHS; see, e.g., Berlinet and Thomas-Agnan 2004; Paulsen and Raghupathi 2016) is a Hilbert space of functions $\mathcal{H}_k$ on a domain $\mathcal{Z}$, equipped with a positive definite kernel $k : \mathcal{Z}\times\mathcal{Z} \to \mathbb{R}$ satisfying the reproducing property:

1. For every $\boldsymbol{z} \in \mathcal{Z}$, the function $k(\boldsymbol{z},\cdot)$ belongs to $\mathcal{H}_k$.
2. For every $f \in \mathcal{H}_k$ and $\boldsymbol{z} \in \mathcal{Z}$, the evaluation functional is continuous and satisfies

$$f(\boldsymbol{z}) = \langle f, k(\boldsymbol{z},\cdot)\rangle_{\mathcal{H}_k}. \tag{2.18}$$

The space $\mathcal{H}_k$ is uniquely determined by its kernel, and every function in the RKHS can be represented (possibly in the limit[3]) as a linear combination of kernel functions. This result is known as the Moore–Aronszajn theorem (Moore, 1935; Aronszajn, 1950).

For vector-valued functions $\boldsymbol{f} : \mathcal{Z} \to \mathbb{R}^m$ with $m > 1$, we use the m-fold product space $\mathcal{H}_k^m$, where each component function $f_i : \mathcal{Z} \to \mathbb{R}$ lies in the scalar RKHS $\mathcal{H}_k$. The reproducing property then holds componentwise. More general extensions using operator-valued kernels exist (Micchelli and Pontil, 2005; Carmeli et al., 2006), but the product-space construction suffices for our purposes.

Projecting functions from $L^2(\mu;\mathbb{R}^m)$ into the reproducing kernel Hilbert product space $\mathcal{H}_k^m$ is beneficial in two complementary ways. From a *function-space perspective*, projecting $\boldsymbol{f}$ into the RKHS $\mathcal{H}_k^m$ enforces smoothness and structure through the kernel, turning the unstructured $L^2(\mu;\mathbb{R}^m)$ geometry into one where similarity and regularity are intrinsic (Schölkopf et al., 2001; Steinwart and Christmann, 2008).

From an *expectation–space perspective*, for $\boldsymbol{f} \in \mathcal{H}_k^m$, expectations become representable and computable through the reproducing property applied componentwise. For a finite signed measure μ on $\mathcal{Z}$ (e.g., a probability distribution Q on $\mathcal{Z}$) each component satisfies

$$\mathbb{E}_\mu[f_i] = \langle f_i, m_\mu\rangle_{\mathcal{H}_k}, \qquad m_\mu(\boldsymbol{z}) = \mathbb{E}_\mu[k(\boldsymbol{z},\cdot)], \quad \text{for } i = 1,\dots,m, \tag{2.19}$$

where m_μ is the kernel mean embedding (Berlinet and Thomas-Agnan, 2004; Gretton et al., 2012) and a kernel k is called *characteristic* if the map $\mu \mapsto m_\mu$ is injective, that is, if $m_{\mu^{(1)}} = m_{\mu^{(2)}}$

[3] Assuming the sum converges in the norm of $\mathcal{H}_k$.

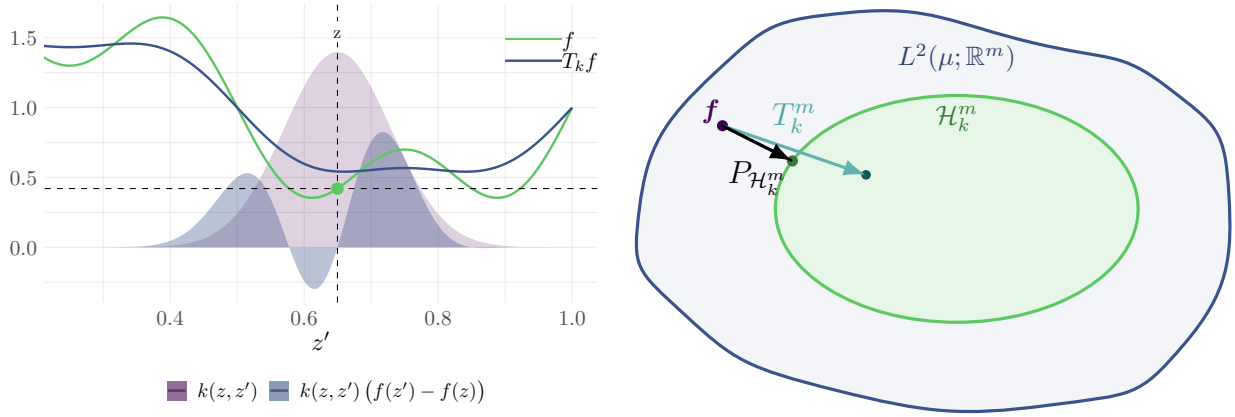

Figure 2.2.: Left: Kernel integral operator visualization $(T_k f)(\boldsymbol{z}) = \int k(\boldsymbol{z}, \boldsymbol{z}') f(\boldsymbol{z}')\, d\mu(\boldsymbol{z}')$, showing kernel-weighted contributions and smoothing bias relative to f. Right: Conceptual geometry of $\mathcal{L}^2(\mu; \mathbb{R}^m)$ and the RKHS $\mathcal{H}_k^m$, indicating the L^2 projection $P_{\mathcal{H}_k^m} f$ to the boundary of $\mathcal{H}_k^m$ and the regularized projection $T_k^m f$ inside $\mathcal{H}_k^m$.

implies $\mu^{(1)} = \mu^{(2)}$ where $\mu^{(1)}$ and $\mu^{(2)}$ are finite signed measures on $\mathcal{Z}$. This identity turns integration into an inner product with the kernel mean, enabling consistent empirical estimates and uniform control of expectations over the RKHS unit ball, i.e.,

$$\|\mathbb{E}_\mu[\boldsymbol{f}]\|_2 \leq \|m_\mu\|_{\mathcal{H}_k} \text{ for all } \boldsymbol{f} \in \mathcal{H}_k^m \text{ with } \|\boldsymbol{f}\|_{\mathcal{H}_k^m} \leq 1 \tag{2.20}$$

since $\|\mathbb{E}_\mu[\boldsymbol{f}]\|_2 \leq \|\boldsymbol{f}\|_{\mathcal{H}_k^m} \|m_\mu\|_{\mathcal{H}_k}$. When k is bounded, $\boldsymbol{z} \mapsto k(\boldsymbol{z}, \cdot)$ has uniformly bounded RKHS norm, so the *Bochner integral* $m_\mu = \int k(\cdot, \boldsymbol{z})\, d\mu(\boldsymbol{z})$ is well defined and satisfies $m_\mu \in \mathcal{H}_k$ (Yosida, 1995; Diestel and Uhl, 1977). This uniform bound yields an integrable dominating function for the Bochner dominated convergence theorem (Diestel and Uhl, 1977), allowing us to interchange gradients and expectations later on.

Together, these properties make projection into $\mathcal{H}_k^m$ both a regularization mechanism for functions and a powerful analytical tool for computing and approximating expectations.

Projecting into the RKHS. In principle, we would like to associate each $\boldsymbol{f} \in L^2(\mu; \mathbb{R}^m)$ with its L^2-orthogonal projection onto $\mathcal{H}_k^m$,

$$P_{\mathcal{H}_k^m} \boldsymbol{f} = \arg\min_{\boldsymbol{g} \in \mathcal{H}_k^m} \|\boldsymbol{f} - \boldsymbol{g}\|^2_{L^2(\mu; \mathbb{R}^m)}. \tag{2.21}$$

However, this projection does not in general exist within $\mathcal{H}_k^m$, since the RKHS need not be closed in $L^2(\mu; \mathbb{R}^m)$ (Berlinet and Thomas-Agnan, 2004; Steinwart and Christmann, 2008) (see the right panel of Figure 2.2). Instead we focus on a regularized projection which is well-defined within $\mathcal{H}_k^m$ given by

$$T_k^m \boldsymbol{f} = \arg\min_{\boldsymbol{g} \in \mathcal{H}_k^m} -\langle \boldsymbol{f}, \boldsymbol{g} \rangle_{L^2(\mu; \mathbb{R}^m)} + \frac{1}{2} \|\boldsymbol{g}\|^2_{\mathcal{H}_k^m}. \tag{2.22}$$

Under the assumption that k is bounded and continuous, this objective has a unique minimizer $T_k^m \boldsymbol{f} \in \mathcal{H}_k^m$ for which it holds that $\langle T_k^m \boldsymbol{f}, \boldsymbol{h} \rangle_{\mathcal{H}_k^m} = \langle \boldsymbol{f}, \boldsymbol{h} \rangle_{L^2(\mu; \mathbb{R}^m)}$ for all $\boldsymbol{h} \in \mathcal{H}_k^m$, i.e., $T_k^m \boldsymbol{f}$ reproduces from $\boldsymbol{f}$ precisely those linear functionals that can be represented by smooth RKHS test

functions $\boldsymbol{h}$ (Berlinet and Thomas-Agnan, 2004; Steinwart and Christmann, 2008). Furthermore, this projection admits an explicit representation:

$$(T_k^m \boldsymbol{f})(\boldsymbol{z}) = \big(T_k f_1(\boldsymbol{z}), \dots, T_k f_m(\boldsymbol{z})\big)^\top \text{ with } T_k f_i = \int_{\mathcal{Z}} k(\cdot, \boldsymbol{z}')\, f_i(\boldsymbol{z}')\, d\mu(\boldsymbol{z}'), \tag{2.23}$$

where each component $T_k f_i \in \mathcal{H}_k$ is a well-defined Bochner integral. Each component $T_k f_i(\boldsymbol{z})$ is obtained by locally averaging f_i around $\boldsymbol{z}$ according to the kernel $k(\boldsymbol{z}, \cdot)$, as illustrated in the left panel of Figure 2.2.

Common choices of bounded continuous kernels such as the Gaussian, Matérn, and inverse multiquadric (IMQ) kernels have a strictly positive spectral density, meaning that the Fourier transform of their translation-invariant form $k(\boldsymbol{z}, \boldsymbol{z}') = \psi(\boldsymbol{z} - \boldsymbol{z}')$ is strictly positive for all frequencies. This property implies both that these kernels are characteristic and that their RKHS is dense in $L^2(\mu; \mathbb{R})$ whenever μ has full support. Consequently, the product space $\mathcal{H}_k^m$ can approximate any function in $L^2(\mu; \mathbb{R}^m)$ arbitrarily well, and the RKHS test functions $\boldsymbol{h} \in \mathcal{H}_k^m$ appearing in the identity

$$\langle T_k^m \boldsymbol{f}, \boldsymbol{h} \rangle_{\mathcal{H}_k^m} = \langle \boldsymbol{f}, \boldsymbol{h} \rangle_{L^2(\mu; \mathbb{R}^m)} \tag{2.24}$$

are expressive enough to probe all $L^2(\mu; \mathbb{R}^m)$-detectable components of $\boldsymbol{f}$.

Functional Derivatives. Let $\mathcal{H}$ be a Hilbert space of vector-valued functions $\boldsymbol{f} \colon \mathcal{Z} \to \mathbb{R}^m$. A functional is a mapping $F \colon \mathcal{H} \to \mathbb{R}$ that assigns a real value to each such function. For $\boldsymbol{f}, \boldsymbol{h} \in \mathcal{H}$, the *Gâteaux derivative* of F at $\boldsymbol{f}$ in the direction $\boldsymbol{h}$ is defined as follows (Deimling, 1985):

$$DF(\boldsymbol{f})[\boldsymbol{h}] = \lim_{t \to 0} \frac{F(\boldsymbol{f} + t\boldsymbol{h}) - F(\boldsymbol{f})}{t}. \tag{2.25}$$

When a functional depends on $\boldsymbol{f}$ only through finitely many pointwise evaluations $\boldsymbol{f}(\boldsymbol{z}^{(i)}) \in \mathbb{R}^m$ for $i = 1, \dots, n$, or through expectations of such evaluations, it can be written as the composition

$$\mathcal{H} \xrightarrow{\boldsymbol{f} \mapsto (\boldsymbol{f}(\boldsymbol{z}^{(1)}), \dots, \boldsymbol{f}(\boldsymbol{z}^{(n)}))} \mathbb{R}^{mn} \xrightarrow{G} \mathbb{R}. \tag{2.26}$$

The chain rule then yields

$$DF(\boldsymbol{f})[\boldsymbol{h}] = \sum_{i=1}^{n} \boldsymbol{a}^{(i)}(\boldsymbol{f})^\top \boldsymbol{h}(\boldsymbol{z}^{(i)}), \tag{2.27}$$

for coefficient vectors $\boldsymbol{a}^{(i)}(\boldsymbol{f}) \in \mathbb{R}^m$. The map $\boldsymbol{h} \mapsto \boldsymbol{h}(\boldsymbol{z}^{(i)})$ is the evaluation functional at $\boldsymbol{z}^{(i)}$, which can be written as integration against the Dirac measure $\delta_{\boldsymbol{z}^{(i)}}$.

This representation shows that the *functional derivative* $DF(\boldsymbol{f})$ is naturally identified with the finite $\mathbb{R}^m$-valued signed measure (Folland, 2013)

$$\mu_{DF(\boldsymbol{f})} = \sum_{i=1}^{n} \boldsymbol{a}^{(i)}(\boldsymbol{f})\, \delta_{\boldsymbol{z}^{(i)}}, \tag{2.28}$$

whose action on a direction $\boldsymbol{h}$ can be written[4] as

$$DF(\boldsymbol{f})[\boldsymbol{h}] = \mathbb{E}_{\mu_{DF(\boldsymbol{f})}}[\boldsymbol{h}(\cdot)^\top] = \sum_{i=1}^{n} \boldsymbol{a}^{(i)}(\boldsymbol{f})^\top \boldsymbol{h}(\boldsymbol{z}^{(i)}). \tag{2.29}$$

[4] Here $\mathbb{E}_{\mu_{DF(\boldsymbol{f})}}[\cdot]$ denotes integration with respect to the vector-valued measure $\mu_{DF(\boldsymbol{f})}$, applied componentwise.

When $\mathcal{H} = L^2(\mu; \mathbb{R}^m)$, point evaluation is not continuous, so $\delta_{\boldsymbol{z}^{(i)}}$ is not a bounded linear functional and $DF(\boldsymbol{f})$ cannot be represented by an element of $L^2(\mu; \mathbb{R}^m)$. In contrast, when $\mathcal{H}$ is an RKHS $\mathcal{H}_k^m$, the Moore–Aronszajn theorem ensures that evaluation is continuous and represented componentwise by the kernel section $k(\boldsymbol{z}^{(i)}, \cdot)$. In this setting, the signed measure (2.28) admits a (componentwise) kernel mean embedding

$$\boldsymbol{m}_{DF(\boldsymbol{f})} = \mathbb{E}_{\mu_{DF(\boldsymbol{f})}}[k(\boldsymbol{z}, \cdot)] = \sum_{i=1}^{n} k(\boldsymbol{z}^{(i)}, \cdot)\, \boldsymbol{a}^{(i)}(\boldsymbol{f}), \tag{2.30}$$

which yields the unique element of $\mathcal{H}_k^m$ representing the action of the functional derivative on RKHS test functions. In particular, for all $\boldsymbol{h} \in \mathcal{H}_k^m$,

$$DF(\boldsymbol{f})[\boldsymbol{h}] = \langle \boldsymbol{m}_{DF(\boldsymbol{f})}, \boldsymbol{h} \rangle_{\mathcal{H}_k^m}. \tag{2.31}$$

Hence, we identify $DF(\boldsymbol{f})$ with its representer $\boldsymbol{m}_{DF(\boldsymbol{f})} \in \mathcal{H}_k^m$. Moreover, Eq. (2.31) implies that the direction of steepest ascent of F in the RKHS geometry is

$$\arg\max_{\boldsymbol{h} \in \mathbb{B}_{\mathcal{H}_k^m}} DF(\boldsymbol{f})[\boldsymbol{h}] = \frac{\boldsymbol{m}_{DF(\boldsymbol{f})}}{\|\boldsymbol{m}_{DF(\boldsymbol{f})}\|_{\mathcal{H}_k^m}}, \quad \text{where } \mathbb{B}_{\mathcal{H}_k^m} := \left\{ \boldsymbol{h} \in \mathcal{H}_k^m \mid \|\boldsymbol{h}\|_{\mathcal{H}_k^m} = 1 \right\}. \tag{2.32}$$

For functionals involving expectations, the identities established earlier for kernel mean embeddings apply: bounded kernels ensure that expectations of RKHS functions exist as Bochner integrals in $\mathcal{H}_k^m$, taken componentwise, and the uniform bound on $\|k(\boldsymbol{z}, \cdot)\|_{\mathcal{H}_k}$ provides the domination required for the Bochner dominated convergence theorem. This justifies interchanging expectations and functional derivatives.

2.3. Foundations of Probabilistic Inference

This chapter introduces the foundational framework of probabilistic inference, which provides a unified representation of randomness and uncertainty across fields such as statistics, machine learning, and physics. By describing quantities of interest in terms of probability distributions, we are able to formalize inference objectives across a broad range of applications.

Bayesian Inference. Bayesian inference updates a so-called *prior distribution* Π of a random *latent* variable $\boldsymbol{z}$ with state space $(\mathcal{Z}, \mathcal{B}_{\mathcal{Z}})$ after observing a realization of the *observable* $\boldsymbol{u}$, which is also treated as a random variable with state space $(\mathcal{U}, \mathcal{B}_{\mathcal{U}})$. The joint distribution of $(\boldsymbol{u}, \boldsymbol{z})$ on $(\mathcal{U} \times \mathcal{Z}, \mathcal{B}_{\mathcal{U}} \otimes \mathcal{B}_{\mathcal{Z}})$[5] is the unique measure Λ that satisfies

$$\Lambda(A_{\mathcal{U}} \times A_{\mathcal{Z}}) = \int_{A_{\mathcal{Z}}} P(A_{\mathcal{U}} \mid \boldsymbol{z})\, \Pi(d\boldsymbol{z}) \qquad \text{for all } A_{\mathcal{U}} \in \mathcal{B}_{\mathcal{U}},\ A_{\mathcal{Z}} \in \mathcal{B}_{\mathcal{Z}}, \tag{2.33}$$

where $P(\cdot \mid \boldsymbol{z})$ is a probability kernel, also called the *observation model*. By construction, $P(\cdot \mid \widetilde{\boldsymbol{z}})$ is the conditional distribution of $\boldsymbol{u}$ given $\boldsymbol{z} = \widetilde{\boldsymbol{z}}$ with respect to the joint distribution Λ, which represents the model assumption of how the data $\boldsymbol{u}$ is generated given $\boldsymbol{z} = \widetilde{\boldsymbol{z}}$. Hence, the prior

[5] $\mathcal{B}_{\mathcal{U}} \otimes \mathcal{B}_{\mathcal{Z}}$ denotes the product σ-algebra.

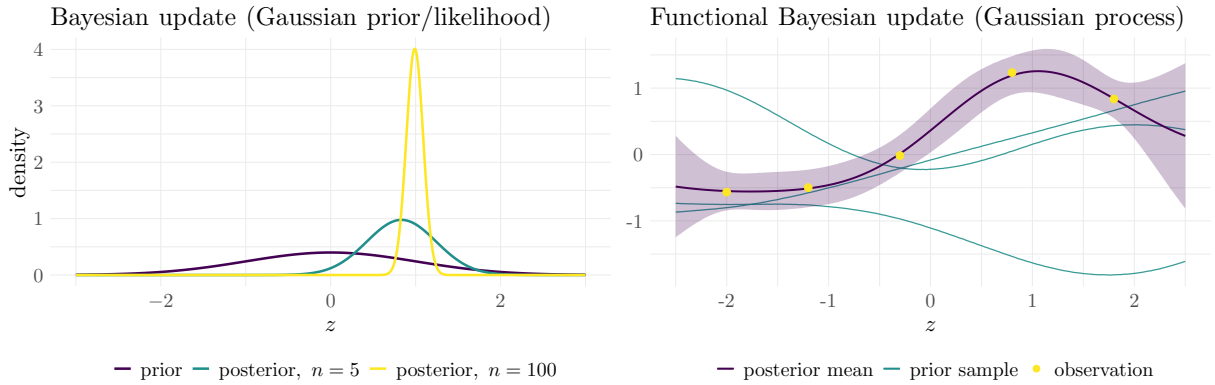

Figure 2.3.: Bayesian updating in both parametric and nonparametric (Gaussian process) models. Left: parametric Gaussian posterior update, showing how the posterior concentrates as the number of observations n increases from $n = 5$ to $n = 100$. Right: Gaussian process prior samples, posterior mean and 2-σ (95%) uncertainty band, and observed data points.

distribution Π indirectly encodes the prior belief about how the observable is distributed, as it induces the (marginal) *prior predictive distribution* of the data given by

$$\Lambda_{\mathcal{U}}(A_{\mathcal{U}}) = \Lambda(A_{\mathcal{U}} \times \mathcal{Z}). \tag{2.34}$$

The *posterior distribution* $\Pi(\cdot \mid \widetilde{\boldsymbol{u}})$ is the conditional distribution of $\boldsymbol{z}$ given $\boldsymbol{u} = \widetilde{\boldsymbol{u}}$, defined by

$$\Lambda(A_{\mathcal{U}} \times A_{\mathcal{Z}}) = \int_{A_{\mathcal{U}}} \Pi(A_{\mathcal{Z}} \mid \boldsymbol{u})\, \Lambda_{\mathcal{U}}(d\boldsymbol{u}) \qquad \text{for all } A_{\mathcal{U}} \in \mathcal{B}_{\mathcal{U}},\ A_{\mathcal{Z}} \in \mathcal{B}_{\mathcal{Z}}. \tag{2.35}$$

The posterior distribution always exists and is uniquely defined $\Lambda_{\mathcal{U}}$-almost surely. It represents the updated belief about $\boldsymbol{z}$ given $\boldsymbol{u} = \widetilde{\boldsymbol{u}}$. It induces the *posterior predictive distribution* of the realized observable $\widetilde{\boldsymbol{u}}$ given by

$$\Lambda_{\mathcal{U}}^{\text{post}}(A_{\mathcal{U}} \mid \widetilde{\boldsymbol{u}}) = \int_{\mathcal{Z}} P(A_{\mathcal{U}} \mid \boldsymbol{z})\, \Pi(d\boldsymbol{z} \mid \widetilde{\boldsymbol{u}}) \qquad \text{for all } A_{\mathcal{U}} \in \mathcal{B}_{\mathcal{U}}. \tag{2.36}$$

While this fully general, measure-theoretic formulation provides a rigorous foundation for Bayesian inference, it is often too abstract for practical computation. To proceed, we typically assume that the posterior is absolutely continuous with respect to the prior, so that a Radon–Nikodym derivative exists (e.g., Schervish, 1995). This can fail, for example, if the prior assigns zero mass to regions where the observable is possible, such as when a discrete prior is used for a continuous parameter. Here, we characterize the *likelihood* as a nonnegative measurable function $L(\cdot \mid \boldsymbol{z})$ proportional to the Radon–Nikodym derivative of the posterior with respect to the prior distribution such that

$$\frac{d\Pi(\cdot \mid \widetilde{\boldsymbol{u}})}{d\Pi}(\boldsymbol{z}) \propto L(\widetilde{\boldsymbol{u}} \mid \boldsymbol{z}). \tag{2.37}$$

This expresses the posterior as a reweighted version of the prior, such that

$$\Pi(A_{\mathcal{Z}} \mid \widetilde{\boldsymbol{u}}) = \frac{\int_{A_{\mathcal{Z}}} L(\widetilde{\boldsymbol{u}} \mid \boldsymbol{z})\, \Pi(d\boldsymbol{z})}{\int_{\mathcal{Z}} L(\widetilde{\boldsymbol{u}} \mid \boldsymbol{z})\, \Pi(d\boldsymbol{z})}. \tag{2.38}$$

This formulation is the foundation for Bayesian inference in function-space settings, where priors are probability measures on infinite-dimensional spaces, such as Gaussian process priors. The right panel of Figure 2.3 illustrates this with a Gaussian process posterior update.

When, in addition, the prior Π itself admits a density $p(\boldsymbol{z})$ with respect to a σ-finite base measure μ (such as Lebesgue measure on $\mathbb{R}^d$), Bayes' rule reduces to the familiar density form

$$p(\boldsymbol{z} \mid \widetilde{\boldsymbol{u}}) = \frac{L(\widetilde{\boldsymbol{u}} \mid \boldsymbol{z})\, p(\boldsymbol{z})}{Z(\widetilde{\boldsymbol{u}})}, \qquad Z(\widetilde{\boldsymbol{u}}) = \int L(\widetilde{\boldsymbol{u}} \mid \boldsymbol{z})\, p(\boldsymbol{z})\, \mu(d\boldsymbol{z}), \tag{2.39}$$

where $p(\cdot \mid \widetilde{\boldsymbol{u}})$ is the density of the posterior distribution $\Pi(\cdot \mid \widetilde{\boldsymbol{u}})$ and the likelihood $L(\cdot \mid \boldsymbol{z})$ is the density of the observation model $P(\cdot \mid \boldsymbol{z})$. This density-based expression is the standard formulation in finite-dimensional statistics and machine learning (see left panel of Figure 2.3).

Consistency of Bayesian Inference. Assume a non-degenerate prior and a true underlying factorized distribution P^*, meaning that the observable $\boldsymbol{u} \in \bigcup_{N\in\mathbb{N}} \widetilde{\mathcal{U}}^N$ is interpreted, conditional on its length $N = \dim(\boldsymbol{u})$, as an N-tuple of independent observations drawn from a probability distribution P_* on a suitable measurable space $\widetilde{\mathcal{U}}$. A central question is whether the posterior predictive distribution $\Lambda_{\mathcal{U}}^{\text{post}}$ is consistent, i.e., whether, as $N = \dim(\boldsymbol{u}) \to \infty$, it weakly converges to the true product distribution $P_*^{\otimes N}$ that defines P^*. When P_* admits a density p_*, such consistency typically holds under standard regularity conditions; this applies to well-specified parametric models and to universal approximators that can grow their capacity with N (Schervish, 1995). In contrast, when P_* does not admit a density, the posterior predictive distribution may fail to converge to the corresponding product measure $P_*^{\otimes N}$ even if the model has infinite capacity (Freedman, 1999). This is a major challenge in nonparametric Bayesian inference.

Bayesian Inference in the Supervised Setting. In the supervised setting, we assume that the data-generating process produces pairs $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) \overset{\text{i.i.d.}}{\sim} P_{\boldsymbol{xy}}$ by first sampling inputs $\boldsymbol{x}^{(i)} \overset{\text{i.i.d.}}{\sim} P_{\boldsymbol{x}}$ on $\mathcal{X}$ and then sampling the corresponding outputs $\boldsymbol{y}^{(i)} \sim P_*(\cdot \mid \boldsymbol{x}^{(i)})$ on $\mathcal{Y}$. Conditional on realized inputs $\widetilde{\boldsymbol{x}}^{(1)}, \ldots, \widetilde{\boldsymbol{x}}^{(N)}$, the observable

$$\boldsymbol{u} = (\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N)}) \tag{2.40}$$

is a random variable taking values in $\mathcal{U} = \bigcup_{N\in\mathbb{N}} \mathcal{Y}^N$, and its components $\boldsymbol{y}^{(i)}$ are assumed conditionally independent given a latent variable $\boldsymbol{z}$.

After observing the corresponding outputs $\widetilde{\boldsymbol{u}} = (\widetilde{\boldsymbol{y}}^{(1)}, \ldots, \widetilde{\boldsymbol{y}}^{(N)})$, the likelihood becomes

$$L(\widetilde{\boldsymbol{u}} \mid \widetilde{\boldsymbol{x}}^{(1)}, \ldots, \widetilde{\boldsymbol{x}}^{(N)}, \boldsymbol{z}) = \prod_{i=1}^{N} l(\widetilde{\boldsymbol{y}}^{(i)} \mid \widetilde{\boldsymbol{x}}^{(i)}, \boldsymbol{z}), \tag{2.41}$$

where $l(\widetilde{\boldsymbol{y}}^{(i)} \mid \widetilde{\boldsymbol{x}}^{(i)}, \boldsymbol{z})$ denotes the likelihood of a single output given its input $\widetilde{\boldsymbol{x}}^{(i)}$ and $\boldsymbol{z}$.

Hence, the density of the posterior predictive distribution for new outputs $\boldsymbol{y}^{*(1)}, \ldots, \boldsymbol{y}^{*(M)}$ given new inputs $\boldsymbol{x}^{*(1)}, \ldots, \boldsymbol{x}^{*(M)}$ is

$$\int_{\mathcal{Z}} \prod_{i=1}^{M} l(\boldsymbol{y}^{*(i)} \mid \boldsymbol{x}^{*(i)}, \boldsymbol{z})\, \Pi(d\boldsymbol{z} \mid \widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{x}}^{(1)}, \ldots, \widetilde{\boldsymbol{x}}^{(N)}). \tag{2.42}$$

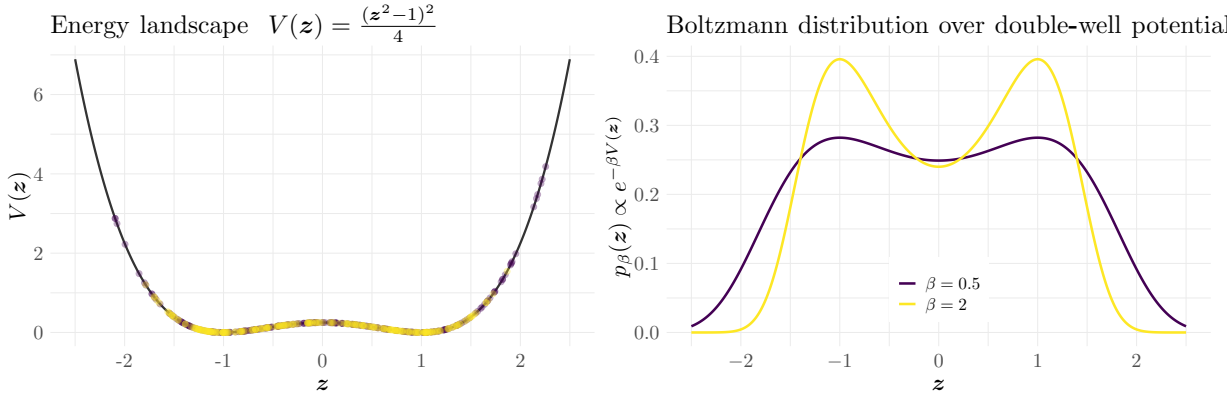

Figure 2.4.: Energy landscape (left) and particle distributions $p_\beta(\boldsymbol{z}) \propto e^{-\beta V(\boldsymbol{z})}$ (right) for $\beta \in \{0.5, 2.0\}$.

Statistical Physics. Continuous probability distributions naturally arise in statistical physics, where they describe the equilibrium[6] states of physical systems (Landau and Lifshitz, 1980; Pathria and Beale, 2022). For a physical system, we model the state as a random variable $\boldsymbol{z}$ taking values in a state space $\mathcal{Z} \subset \mathbb{R}^d$, for example encoding the position and velocity of a particle. The energy of the system is a function $V : \mathcal{Z} \to \mathbb{R}$, and the probability density of its associated Boltzmann distribution at inverse temperature $\beta = 1/(k_B T)$ is given by

$$p(\boldsymbol{z}) = \frac{1}{Z_\beta} \exp\big(-\beta\, V(\boldsymbol{z})\big), \qquad Z_\beta = \int_{\mathcal{Z}} \exp\big(-\beta\, V(\boldsymbol{z})\big)\, d\boldsymbol{z}, \tag{2.43}$$

where k_B is Boltzmann's constant and T is absolute temperature. The energy function V assigns lower values to more probable configurations, since the Boltzmann weight $\exp(-\beta\, V(\boldsymbol{z}))$ suppresses high-energy states and favors low-energy ones, with larger β (lower temperature) yielding a sharper concentration around low-energy configurations (see Figure 2.4). This perspective is central not only in physics but also in applications such as molecular modeling and drug discovery, where equilibrium distributions capture the most likely molecular conformations or protein-ligand binding poses (Zheng et al., 2024). However, direct sampling via simulation (e.g., molecular dynamics or Monte Carlo) is often prohibitively expensive in high dimensions. Variational inference offers an attractive alternative: it aims to approximate these equilibrium distributions in a fast, amortized manner, potentially enabling the efficient generation of high-quality samples without costly simulation.

Curse of Dimensionality. As the dimension d of the finite-dimensional real-valued state space $\mathcal{Z} \subset \mathbb{R}^d$ increases, probabilistic inference becomes severely more difficult. This phenomenon, widely termed the *curse of dimensionality*, strongly affects the efficiency and reliability of inference procedures. For large d, distributions tend to concentrate their probability mass on thin, low-volume regions[7] called the *typical set* (Betancourt, 2018). This is challenging for inference algorithms, which must locate and explore the high-dimensional typical set. As d increases, the typical set also develops highly structured geometry, such as the thin-shell concentration seen for Gaussian distributions (see the left panel of Figure 2.5). A crucial consequence is that small perturbations in mean, scale, or covariance produce a large reduction in the overlap between the

[6] The state in which complex many-particle systems become predictable at the macroscopic level.
[7] This effect is known as the concentration of measure phenomenon (Ledoux, 2001).

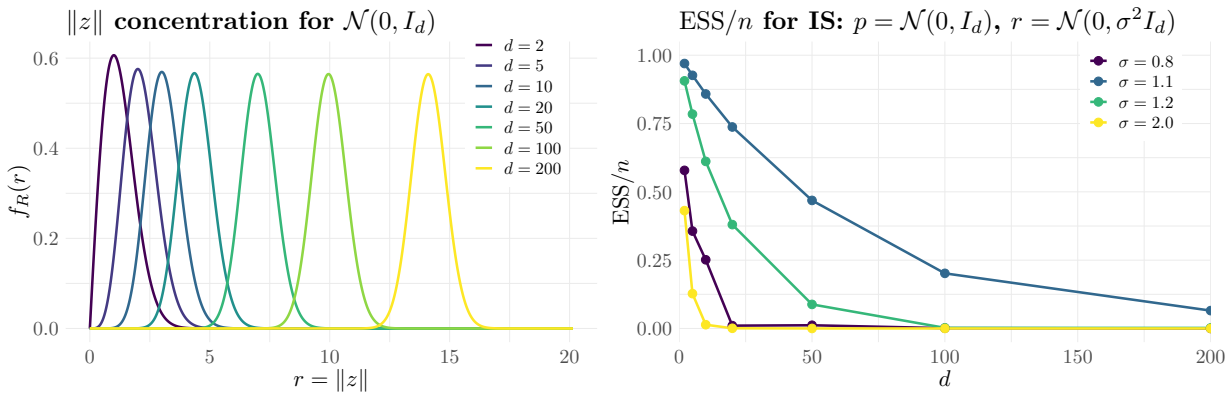

Figure 2.5.: Curse of dimensionality (CoD) illustrations. Left: Concentration of the radius $\|z\|$ for $z \sim \mathcal{N}(0, I_d)$ in a thin shell as d grows. Right: Importance sampling (IS) effective sample size fraction ESS/n deteriorates with dimension when the proposal $r = \mathcal{N}(0, \sigma^2 I_d)$ is scale-mismatched from the target $p = \mathcal{N}(0, I_d)$; curves shown for several σ values.

typical sets of two distributions, which in variational inference and importance sampling can lead to severe importance-weight degeneracy (Agapiou et al., 2017). This degeneracy leads to a poor effective sample size (ESS), as illustrated in the right panel of Figure 2.5. Hence, naive (uninformed) sampling schemes scale poorly with dimension, motivating the development of specialized Monte Carlo and VI methods.

2.4. Comparing Distributions via Divergences

In variational inference, we are interested in approximating a target distribution P given by a (possibly unnormalized) probability density function p with a distribution Q that is feasible to sample from. However, in order to approximate distributions, we first need a measure of how different two distributions are. We call this measure $\mathrm{D}(Q\|P)$ the *divergence* between Q and P.

A divergence D, which maps two probability measures to a real number, is characterized by:

1. **Non-negativity:** $\mathrm{D}(Q\|P) \geq 0$ for all Q and P.
2. **Definiteness:** $\mathrm{D}(Q\|P) = 0$ if and only if $Q = P$.

These properties ensure that the divergence is a valid measure of discrepancy between distributions. A divergence represents the most relaxed but still meaningful notion of a distance on probability measures: a divergence does not have to satisfy the triangle inequality or symmetry. However, every metric on probability measures is also a divergence.

A natural way to distinguish divergences is by examining the mathematical objects associated with the target distribution P that appear in their *definitions*. This yields three main classes:

1. **Density-based divergences:** Their definition makes explicit use of the density p, but without involving gradients.
2. **Score-based divergences:** Their definition involves the score $\boldsymbol{s}_p = \nabla_{\boldsymbol{z}} \log p$, i.e., the gradient of the log density with respect to $\boldsymbol{z}$.

3. **Sample-based divergences:** Their definition is formulated directly in terms of the underlying probability measures P and Q, without involving densities or scores.

In variational inference, since we usually do not have access to samples from the target P, we favor divergences that can be estimated with samples from the approximation Q, together with evaluations of $\log p$ or its score $\nabla_{\boldsymbol{z}} \log p$. Although sample-based divergences are rarely used as objective functions in VI, they remain valuable as evaluation tools, particularly in benchmark or diagnostic settings where draws from P are available.

Another important distinction is whether a divergence tends to promote *mode seeking* or *mass covering* approximations. Mode-seeking divergences focus on the modes already captured by the approximation, sharpening local high-density regions. In contrast, mass-covering divergences penalize missing probability mass with respect to the target, pushing the approximation to represent all regions where the target distribution has non-negligible probability mass. In this work, we are primarily interested in divergences that are appropriate for targets whose high-density regions are not separated by large low-density regions, rather than divergences that explicitly encourage broad exploration or mass-covering behavior.

2.4.1. Density-Based Divergences

Density-based divergences are defined explicitly in terms of the density p of the target distribution P, without involving gradients. The most well-known example is the Kullback–Leibler divergence, which has historically been the starting point for modern variational inference methods (Jordan et al., 1998). Since the main results in this work are based on the KL divergence, we will discuss it in detail, while also presenting other density-based divergences that are related to the KL.

Kullback–Leibler (KL) Divergence. The KL divergence is fundamental not only in statistics but also in *information theory*, where it characterizes the inefficiency or "extra" number of nats incurred when using Q to encode data generated from P (MacKay, 2003). Remarkably, the KL divergence is *uniquely* determined (up to a constant factor) as the only measure of discrepancy between probability distributions that satisfies the following three properties:

(i) **Non-negativity and definiteness:** $\mathrm{D}_{\mathrm{KL}}(Q \,\|\, P) \geq 0$, with equality if and only if $Q = P$.

(ii) **Additivity under independence:** If $Q = Q_1 \otimes Q_2$ and $P = P_1 \otimes P_2$ (i.e., Q and P are product measures over their respective marginals Q_1, Q_2 and P_1, P_2), then

$$\mathrm{D}_{\mathrm{KL}}(Q_1 \otimes Q_2 \,\|\, P_1 \otimes P_2) = \mathrm{D}_{\mathrm{KL}}(Q_1 \,\|\, P_1) + \mathrm{D}_{\mathrm{KL}}(Q_2 \,\|\, P_2). \tag{2.44}$$

This property ensures that D_{KL} measures total discrepancy additively across independent components, without introducing artificial dependencies.

(iii) **Data processing (Markov) property:** For any measurable map T, let $T_{\#}Q$ and $T_{\#}P$ denote the pushforward measures of Q and P under T. Then

$$\mathrm{D}_{\mathrm{KL}}(T_{\#}Q \,\|\, T_{\#}P) \leq \mathrm{D}_{\mathrm{KL}}(Q \,\|\, P). \tag{2.45}$$

This property guarantees that no mapping can artificially increase the perceived difference between distributions. This has important implications for VI as discussed in Section 2.4.4.

Table 2.1.: Examples of f-divergences for typical choices of f, with induced support sensitivity.

$f(t)$	Divergence Name	Expression	Support Sensitivity
$t \log t$	Reverse KL	$\mathrm{D}_{\mathrm{KL}}(q \,\|\, p)$	Mode-seeking
$-\log t$	Forward KL	$\mathrm{D}_{\mathrm{KL}}(p \,\|\, q)$	Mass-covering
$(\sqrt{t}-1)^2$	$2\times$ Squared Hellinger	$2H^2(q,p)$	Balanced
$(t-1)^2$	χ^2 divergence	$\mathrm{D}_{\chi^2}(q \,\|\, p)$	Mode-seeking
$\frac{1}{2}t \log \frac{2t}{1+t} + \frac{1}{2} \log \frac{2}{1+t}$	Jensen-Shannon (JS)	$\mathrm{D}_{\mathrm{JS}}(q,p)$	Balanced

As noted above and formalized by Csiszár (1967), any divergence satisfying these three properties is necessarily a constant multiple of the KL divergence, which is given by

$$\mathrm{D}_{\mathrm{KL}}(Q \,\|\, P) := \int_{\mathcal{Z}} \log\left(\frac{dQ}{dP}(\boldsymbol{z})\right) Q(d\boldsymbol{z}), \tag{2.46}$$

for probability measures Q and P on $(\mathcal{Z}, \mathcal{B})$ where Q is absolutely continuous with respect to P. Conversely, no other divergence satisfies all three of these properties. This uniqueness explains the centrality of the KL divergence in information theory and variational inference.

If both Q and P admit densities q and p w.r.t. Lebesgue measure λ respectively,

$$\mathrm{D}_{\mathrm{KL}}(q \,\|\, p) = \int_{\mathcal{Z}} q(\boldsymbol{z}) \log\left(\frac{q(\boldsymbol{z})}{p(\boldsymbol{z})}\right) \lambda(d\boldsymbol{z}). \tag{2.47}$$

f-Divergences. Let $f : (0, \infty) \to \mathbb{R}$ be convex with $f(1) = 0$. The Csiszár–Morimoto f-divergence between densities q and p is

$$\mathrm{D}_f(q \,\|\, p) := \int_{\mathcal{Z}} p(\boldsymbol{z})\, f\left(\frac{q(\boldsymbol{z})}{p(\boldsymbol{z})}\right) \lambda(d\boldsymbol{z}). \tag{2.48}$$

This family encompasses a wide variety of divergences. Table 2.1 presents representative examples (here we follow the VI convention for naming forward and reverse KL), where each specific divergence is determined by the particular choice of convex function f.

Csiszár 1967 showed that f-divergences are the only divergences that satisfy the data processing (Markov) property. There are many proposed VI approaches based on f-divergences. However, the reverse KL divergence remains central in VI because it is the only f-divergence that can be estimated via MC without bias (up to an additive constant) when the target distribution is not normalized and we cannot sample from P which is the general VI use case (see Section 2.5.2). However, the reverse KL is also known to be mode-seeking, which is not always desirable in VI.

2.4.2. Score-Based Divergences

Score-based discrepancies provide an alternative to density-based divergences for comparing distributions, quantifying differences in terms of the scores, i.e., the gradients of the log densities. We use the word "discrepancy" to clarify that, in general, they are not divergences in the strict

sense of the word, and they need specific design choices to become a divergence. Several modern VI approaches make use of score-based divergences as core components. They also are a major factor in recent advances in generative modeling, but our focus here is on their utility for VI.

The general form for a score-based discrepancy is as follows: given the score functions $\boldsymbol{s}_p(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z})$ and $\boldsymbol{s}_q(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \log q(\boldsymbol{z})$, the discrepancy takes the form

$$\mathrm{D}_{\mathcal{F}}(Q \,\|\, P) := \sup_{\boldsymbol{f} \in \mathcal{F}} \mathbb{E}_{\boldsymbol{z} \sim Q}\big[(\boldsymbol{s}_p(\boldsymbol{z}) - \boldsymbol{s}_q(\boldsymbol{z}))^\top \boldsymbol{f}(\boldsymbol{z})\big], \tag{2.49}$$

where $\mathcal{F}$ is a space of test functions. The particular choice of $\mathcal{F}$, as well as whether the score of q is available, leads to different score-based divergences found in the literature.

Fisher Divergence. For $\mathcal{F} = \{\boldsymbol{f} : \|\boldsymbol{f}\|_{L^2(Q;\mathbb{R}^d)} \leq 1\}$, the score-based discrepancy reduces to the Fisher divergence (also known as score matching (Hyvärinen, 2005)), which is given by

$$\mathrm{D}_{\mathrm{Fisher}}(Q \,\|\, P) = \tfrac{1}{2} \|\boldsymbol{s}_p - \boldsymbol{s}_q\|^2_{L^2(Q;\mathbb{R}^d)} = \mathbb{E}_{\boldsymbol{z} \sim Q}\Big[\tfrac{1}{2} \|\nabla_{\boldsymbol{z}} \log q(\boldsymbol{z}) - \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z})\|^2\Big]. \tag{2.50}$$

This quantity is zero if and only if $\boldsymbol{s}_p(\boldsymbol{z}) = \boldsymbol{s}_q(\boldsymbol{z})$ wherever $q(\boldsymbol{z}) > 0$, and under mild regularity, this implies $q = p$. The maximizer is $\boldsymbol{f}^*(\boldsymbol{z}) \propto \boldsymbol{s}_p(\boldsymbol{z}) - \boldsymbol{s}_q(\boldsymbol{z})$, showing that the Fisher divergence measures the $L^2(Q;\mathbb{R}^d)$-norm of the score mismatch.

Stein Discrepancy. When $\boldsymbol{s}_q$ is unavailable, Stein's identity offers a useful alternative. Using integration by parts under suitable boundary and regularity conditions (e.g., as discussed in Liu and Wang (2016)), one obtains for any test function $\boldsymbol{f}$,

$$\mathbb{E}_{\boldsymbol{z} \sim P}[\mathcal{A}_p \boldsymbol{f}(\boldsymbol{z})] = 0, \qquad \mathcal{A}_p \boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{s}_p(\boldsymbol{z})^\top \boldsymbol{f}(\boldsymbol{z}) + \nabla \cdot \boldsymbol{f}(\boldsymbol{z}), \tag{2.51}$$

where $\mathcal{A}_p$ is the Stein operator and $\nabla \cdot \boldsymbol{f}(\boldsymbol{z})$ is the divergence of $\boldsymbol{f}(\boldsymbol{z})$. Applying this identity twice allows us to express the expectation of $\mathcal{A}_p$ under Q as

$$\mathbb{E}_{\boldsymbol{z} \sim Q}[\mathcal{A}_p \boldsymbol{f}(\boldsymbol{z})] = \mathbb{E}_{\boldsymbol{z} \sim Q}\big[(\boldsymbol{s}_p(\boldsymbol{z}) - \boldsymbol{s}_q(\boldsymbol{z}))^\top \boldsymbol{f}(\boldsymbol{z})\big], \tag{2.52}$$

which matches the inner term of the general score-based discrepancy form. So the general Stein discrepancy can be written as

$$\mathrm{D}_{\mathrm{Stein},\mathcal{F}}(Q \| P) := \sup_{\boldsymbol{f} \in \mathcal{F}} \mathbb{E}_{\boldsymbol{z} \sim Q}[\mathcal{A}_p \boldsymbol{f}(\boldsymbol{z})] \tag{2.53}$$

which no longer depends on $\boldsymbol{s}_q$.

Kernelized Score-Based Discrepancies. When $\mathcal{F} = \{\boldsymbol{f} \in \mathcal{H}_k^d : \|\boldsymbol{f}\|_{\mathcal{H}_k^d} \leq 1\}$ where $\mathcal{H}_k^d$ is a vector-valued RKHS with kernel function k, two forms of kernelized score-based discrepancies can be derived depending on whether Stein's identity is applied.

If $\boldsymbol{s}_q$ is available, the *kernelized Fisher divergence* can be written as

$$\mathrm{D}_{\mathrm{KF}}(Q \| P) := \sup_{\|\boldsymbol{f}\|_{\mathcal{H}_k^d} \leq 1} \mathbb{E}_{\boldsymbol{z} \sim Q}\big[(\boldsymbol{s}_p(\boldsymbol{z}) - \boldsymbol{s}_q(\boldsymbol{z}))^\top \boldsymbol{f}(\boldsymbol{z})\big]. \tag{2.54}$$

Because the objective is linear in $\boldsymbol{f}$ within $\mathcal{H}_k^d$, the supremum has a maximizer $\boldsymbol{f}^*$ in closed form,

$$\boldsymbol{f}^*(\cdot) \propto \mathbb{E}_{\boldsymbol{z}\sim Q}[k(\boldsymbol{z},\cdot)\,(\boldsymbol{s}_p(\boldsymbol{z}) - \boldsymbol{s}_q(\boldsymbol{z}))] \text{ and thus } \mathrm{D}_{\mathrm{KF}}(Q\|P) = \left\|\mathbb{E}_{\boldsymbol{z}\sim Q}[k(\boldsymbol{z},\cdot)\,(\boldsymbol{s}_p(\boldsymbol{z}) - \boldsymbol{s}_q(\boldsymbol{z}))]\right\|_{\mathcal{H}_k^d}. \tag{2.55}$$

If $\boldsymbol{s}_q$ is not available, under suitable boundary and regularity conditions, Stein's identity can be used to replace it, yielding the *kernelized Stein discrepancy* (KSD) as

$$\mathrm{D}_{\mathrm{KSD}}(Q\|P) := \left\|\boldsymbol{f}^*\right\|_{\mathcal{H}_k^d} \text{ with } \boldsymbol{f}^*(\cdot) \propto \mathbb{E}_{\boldsymbol{z}\sim Q}\left[\nabla_{\boldsymbol{z}} k(\boldsymbol{z},\cdot) + k(\boldsymbol{z},\cdot)\,\boldsymbol{s}_p(\boldsymbol{z})\right]. \tag{2.56}$$

Thus, whether expressed in the both-scores form or in the Steinized form, kernelized score-based discrepancies uniquely combine theoretical tractability with a closed-form optimal direction corresponding to the steepest deviation of Q from P. The $\mathrm{D}_{\mathrm{KSD}}$ generally requires a P-distinguishing kernel (e.g., the IMQ kernel) in order to define a strict divergence, which imposes additional tail conditions (Liu et al., 2016; Chwialkowski et al., 2016; Gorham and Mackey, 2017).

2.4.3. Sample-Based Divergences

Sample-based metrics are an important class of divergences in generative modeling, since they are robust to support mismatch between Q and P. In such cases, divergences like KL and f-divergences may become infinite or undefined. However, for VI we usually assume that Q and P have overlapping non-negligible supports. Below, the two most commonly used sample-based divergences are summarized, which are often employed in VI for evaluating the quality of the approximate distribution Q.

Maximum Mean Discrepancy (MMD). In a scalar-valued RKHS $\mathcal{H}_k$ with kernel function k, the maximum mean discrepancy (Gretton et al., 2012) is defined as

$$\begin{aligned}\mathrm{D}^2_{\mathrm{MMD}}(Q,P) &:= \left\|\mathbb{E}_{\boldsymbol{z}\sim Q}[k(\boldsymbol{z},\cdot)] - \mathbb{E}_{\widetilde{\boldsymbol{z}}\sim P}[k(\widetilde{\boldsymbol{z}},\cdot)]\right\|^2_{\mathcal{H}_k} \\ &= \mathbb{E}_{\boldsymbol{z},\boldsymbol{z}'\sim Q}[k(\boldsymbol{z},\boldsymbol{z}')] + \mathbb{E}_{\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}'\sim P}[k(\widetilde{\boldsymbol{z}},\widetilde{\boldsymbol{z}}')] - 2\,\mathbb{E}_{\boldsymbol{z}\sim Q,\widetilde{\boldsymbol{z}}\sim P}[k(\boldsymbol{z},\widetilde{\boldsymbol{z}})].\end{aligned} \tag{2.57}$$

The kernel mean embedding $\mu_Q := \mathbb{E}_{\boldsymbol{z}\sim Q}[k(\boldsymbol{z},\cdot)]$ encodes all information needed to compute expectations of all functions in $\mathcal{H}_k$ under Q (see Section 2.2). If k is characteristic (e.g., Gaussian RBF), then $\mathrm{D}_{\mathrm{MMD}}(Q,P)$ is a strict divergence.

Wasserstein Distance. On a metric space $(\mathcal{Z},d)$, let $\Pi(Q,P)$ denote the set of all joint distributions on $\mathcal{Z}\times\mathcal{Z}$ with marginal distributions Q and P. The Wasserstein distance (Villani, 2009; Arjovsky et al., 2017) is defined as

$$\mathrm{D}_{W_p}(Q,P) := \left(\inf_{\pi\in\Pi(Q,P)} \int_{\mathcal{Z}\times\mathcal{Z}} d(z,z')^p\,\pi(dz,dz')\right)^{1/p}, \tag{2.58}$$

where $d:\mathcal{Z}\times\mathcal{Z}\to[0,\infty)$ is a metric on $\mathcal{Z}$.

The Wasserstein distance is uniquely sensitive to the geometry of the underlying metric space $(\mathcal{Z},d)$, as it measures the minimum cost of transporting probability mass between Q and P.

2.4.4. From Divergences to Modes of Convergence

So far, we have discussed divergences as static measures of discrepancy between two distributions Q and P. However, in variational inference the situation is inherently dynamic, as we construct a sequence of approximations $Q^{[t]}$ and seek to minimize a divergence $\mathrm{D}(Q^{[t]}\|P)$. Naturally, we say that $Q^{[t]}$ converges to P in divergence if, for every $\varepsilon > 0$, there exists an $M_\varepsilon \in \mathbb{N}$ such that

$$t \geq M_\varepsilon \;\Rightarrow\; \mathrm{D}(Q^{[t]}\|P) < \varepsilon. \tag{2.59}$$

In other words, convergence in divergence means that the sequence eventually remains inside every divergence ball $\{Q : \mathrm{D}(Q\|P) < \varepsilon\}$. Different divergences induce different neighborhood structures around P, and the condition $\mathrm{D}(Q^{[t]}\|P) \to 0$ may correspond to fundamentally different modes of convergence even if the unique minimizer is the same.

Total Variation (Strong Convergence). A sequence $Q^{[t]}$ converges to P in total variation if

$$\|Q^{[t]} - P\|_{\mathrm{TV}} = \sup_{A \in \mathcal{B}} |Q^{[t]}(A) - P(A)| \;\longrightarrow\; 0, \tag{2.60}$$

where $\mathcal{B}$ is the σ-algebra on $\mathcal{Z}$. If $Q^{[t]}$ and P admit densities $q^{[t]}$ and p with respect to a common base measure μ, this becomes

$$\|Q^{[t]} - P\|_{\mathrm{TV}} = \tfrac{1}{2} \int_{\mathcal{Z}} |q^{[t]}(\boldsymbol{z}) - p(\boldsymbol{z})| \, d\mu(\boldsymbol{z}). \tag{2.61}$$

Convergence in total variation is a *strong* mode of convergence: it controls the maximum discrepancy in mass assigned to any measurable set. As a consequence, for every $\boldsymbol{f} \in L^1(P; \mathbb{R}^m)$ (in particular, all finite moments),

$$\mathbb{E}_{\boldsymbol{z} \sim Q^{[t]}}[\boldsymbol{f}(\boldsymbol{z})] \;\longrightarrow\; \mathbb{E}_{\boldsymbol{z} \sim P}[\boldsymbol{f}(\boldsymbol{z})]. \tag{2.62}$$

When densities exist, total variation convergence is equivalent to L^1-convergence of the densities, although this does not, in general, imply pointwise almost-everywhere convergence.

Weak Convergence. A sequence $Q^{[t]}$ converges *weakly* to P if

$$\mathbb{E}_{\boldsymbol{z} \sim Q^{[t]}}[\boldsymbol{f}(\boldsymbol{z})] \;\longrightarrow\; \mathbb{E}_{\boldsymbol{z} \sim P}[\boldsymbol{f}(\boldsymbol{z})] \quad \text{for all bounded continuous functions } \boldsymbol{f} : \mathcal{Z} \to \mathbb{R}^m. \tag{2.63}$$

This is a strictly weaker requirement than convergence in total variation. In particular, weak convergence does not, in general, guarantee convergence of expectations of unbounded functions (such as moments). Every sequence of distributions $\{Q^{[t]}\}_{t=1}^{\infty}$ that converges in total variation converges weakly, but not vice versa.

Partial Hierarchies of Divergences

The following three hierarchies summarize key relationships between divergences and the modes of convergence they induce. Throughout, $A \Rightarrow B$ means that convergence in A implies convergence in B, while $A \Longleftrightarrow B$ means that the two are equivalent under the conditions listed in the corresponding hierarchy.

(1) Information-Theoretic Hierarchy.

$$\begin{array}{ccccc} \text{KL divergence} & \Rightarrow & \text{Total variation} & \Rightarrow & \text{Weak convergence} \\ & & \text{Fisher divergence} & \Rightarrow & \text{Weak convergence} \end{array} \tag{2.64}$$

Pinsker's inequality (Pinsker, 1964; Cover and Thomas, 2006) ensures that $\mathrm{KL}(Q^{[t]} \,\|\, P) \geq 2\,\mathrm{TV}^2(Q^{[t]}, P)$ whenever both sides are finite. The KL and Fisher divergences are not comparable in general; without additional structural assumptions such as a log-Sobolev inequality (Vempala and Wibisono, 2019), Fisher does not control KL or total variation.

(2) Kernel-Based Hierarchy.

$$\begin{array}{rcl} \mathrm{D}_{\mathrm{MMD}} & \iff & \text{Weak convergence,} \\ \mathrm{D}_{\mathrm{KSD}} & \Rightarrow & \text{Weak convergence only for specific kernels and targets.} \end{array} \tag{2.65}$$

For MMD, the equivalence holds precisely when the kernel is *characteristic*, meaning that the associated RKHS is rich enough to distinguish probability measures. For KSD, the situation is more nuanced: the discrepancy controls weak convergence only under joint conditions on the kernel and the target distribution P. Heavy-tailed kernels such as the inverse multiquadratic (IMQ) can ensure that $\mathrm{D}_{\mathrm{KSD}}(Q^{[t]}, P) \to 0$ implies $Q^{[t]} \Rightarrow P$ for certain classes of targets with appropriate regularity and tail behavior, but this implication does not hold in general (Gorham and Mackey, 2017).

(3) Geometric Hierarchy. For $i, j \in \mathbb{N}$ with $i > j$,

$$\mathrm{D}_{W_i} \;\Rightarrow\; \mathrm{D}_{W_j} \;\Rightarrow\; \text{Weak convergence.} \tag{2.66}$$

Convergence in the Wasserstein distance of order i implies convergence in all lower orders $j < i$, and implies convergence of all moments of order up to i, provided these moments are finite for P. All Wasserstein distances induce weak convergence, provided the corresponding finite-moment conditions for P are satisfied (Villani, 2009).

2.5. Variational Inference

In variational inference, we frame inference as an optimization problem (Jordan et al., 1998; Blei et al., 2017); thus we define a family of candidate distributions $\mathcal{Q}$ and seek the member Q^* that best approximates the target distribution P such that

$$Q^* \in \operatorname*{arg\,min}_{Q \in \mathcal{Q}} \; \mathcal{L}(Q \,\|\, P), \tag{2.67}$$

where $\mathcal{L}$ is a loss function, usually based on a divergence between Q and P. Since VI is an optimization problem, different approaches are best described by their respective algorithms.

A VI algorithm $\mathcal{I}_{\boldsymbol{\lambda}}$ with hyperparameters $\boldsymbol{\lambda}$ usually consists of the following three components:

- **Variational family $\mathcal{Q}$:** A tractable family of possible approximating distributions, which may be parametric (e.g., $\{Q_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$), defined by a generative process, or more general.

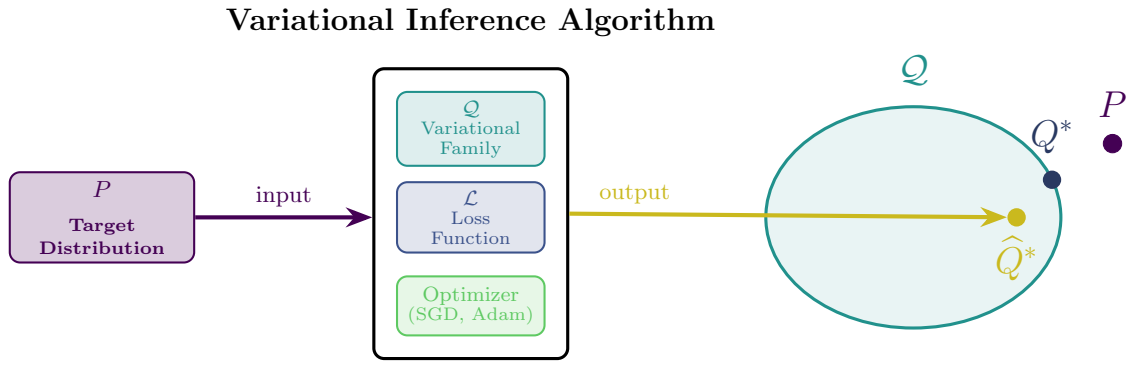

Figure 2.6.: The target distribution P (input, left) given by its unnormalized density p is processed by the VI algorithm consisting of the variational family $\mathcal{Q}$, loss function $\mathcal{L}$, and optimizer. The algorithm outputs $\widehat{Q}^*$ (yellow point), which lies inside $\mathcal{Q}$. The optimal Q^* (dark blue) is at the boundary of $\mathcal{Q}$, closest (as measured with respect to the loss $\mathcal{L}$) to the true target P (purple, right), which lies in general outside $\mathcal{Q}$.

- **Loss function $\mathcal{L}$:** Measures the discrepancy between Q and P (often a divergence). In practice, the exact divergence is often hard to optimize, so a surrogate objective is adopted, usually estimated stochastically (e.g., via samples), and depends on the hyperparameters $\boldsymbol{\lambda}$ and the structure of $\mathcal{Q}$.
- **Optimizer:** A predefined rule for constructing $Q^{[t+1]}$ from $Q^{[t]}$ using a stochastic estimate of $\mathcal{L}$, which is designed to reduce the estimated objective within the variational family $\mathcal{Q}$.

Hence, a VI algorithm is a function $\mathcal{I}_{\boldsymbol{\lambda}}$ that maps a target distribution P, usually given by its unnormalized density p, to an approximate distribution $\widehat{Q}^*$ in the variational family $\mathcal{Q}$ by iteratively applying the optimizer to the loss estimate of $\mathcal{L}(Q^{[t]} \,\|\, P)$. We refer to the discrepancies between $\widehat{Q}^*$ and Q^*, and between Q^* and P, as the *variational gap* and the *approximation gap*, respectively (see Figure 2.6). For many modern VI algorithms, the approximation gap can often be made arbitrarily small but the real challenge is closing the variational gap. This is also strongly influenced by the hyperparameters $\boldsymbol{\lambda}$, which include, e.g., the learning rate, the number of iterations, the neural network architecture, and the batch size. We note the similarity to ML learners (Bischl et al., 2023) and will discuss this in more detail in Section 2.6.

2.5.1. Gradient-Based Optimization in VI

In high-dimensional models, probability mass concentrates on thin typical sets (see Section 2.3), which makes optimization challenging for methods that lack directional information. Gradient-based approaches naturally mitigate this issue, since gradients provide structured signals that guide the optimization toward regions of locally high probability mass without requiring an exhaustive exploration. Variational families $\mathcal{Q}$ that admit efficient gradient-based training procedures can therefore be optimized reliably at scale, which is one of the main reasons why gradient-based methods have become the standard choice in modern VI.

Formally, we now consider a parameterized family of approximating distributions $\mathcal{Q} = \{Q_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$, where $\Theta \subseteq \mathbb{R}^d$ denotes the parameter space. We assume that the mapping $\boldsymbol{\theta} \mapsto q_{\boldsymbol{\theta}}(\boldsymbol{z})$ is differentiable for every $\boldsymbol{z} \in \mathcal{Z}$, ensuring that gradients with respect to $\boldsymbol{\theta}$ are well defined.

In practice, gradients with respect to $\boldsymbol{\theta}$ are efficiently computable in modern deep learning frameworks, whereas higher-order derivatives remain substantially more expensive. Hence, first-order gradient-based methods are commonly used for VI in high-dimensional settings. Gradient descent and other first-order variants such as Adam (Kingma and Ba, 2015) and RMSprop (Hinton et al., 2012) update the parameter vector $\boldsymbol{\theta}$ via an iterative rule,

$$\boldsymbol{\theta}^{[t+1]} = \boldsymbol{\theta}^{[t]} - \eta^{[t]}\, \boldsymbol{g}^{[t]}, \tag{2.68}$$

where $\boldsymbol{g}^{[t]}$ is a first-order search direction (typically constructed from the gradient[8] $\boldsymbol{d}^{[t]} = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{[t]})$) and $\eta^{[t]}$ is the learning rate (step size) at iteration t.

For modern VI approaches, $\mathcal{Q}$ is often highly overparameterized, leading to a strongly non-convex optimization landscape. Although traditional guarantees for gradient-based optimization, such as convergence to a global minimum, rely on the convexity of the loss surface $\mathcal{L}(\boldsymbol{\theta})$, several of the optimization benefits observed in overparameterized deep learning models carry over to VI, smoothing the parameter landscape and creating wide basins of attraction in which first-order methods can make reliable progress (Li et al., 2022; Allen-Zhu et al., 2019). However, the variational objective is typically more complex than standard predictive losses and is prone to suboptimal local minima arising from the optimization incentives induced by different divergences. Mass-covering divergences reward assigning probability mass broadly, which can lead to inflated variance, whereas mode-seeking divergences penalize assigning mass to low-density regions and therefore tend to ignore parts of the target distribution (Minka, 2005; Li and Turner, 2016).

This challenging optimization is further complicated by the fact that in general we do not have access to the gradient $\boldsymbol{d}^{[t]} = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{[t]})$ but only to a stochastic estimate $\widehat{\boldsymbol{d}}^{[t]}$. Stochastic variants of the discussed gradient-based methods can still be successfully applied in this setting, but this relies on the assumption that the mean squared error (MSE) of the stochastic estimate $\widehat{\boldsymbol{d}}^{[t]}$

$$\mathrm{MSE}(\widehat{\boldsymbol{d}}^{[t]}) := \mathbb{E}\left[\left\|\widehat{\boldsymbol{d}}^{[t]} - \boldsymbol{d}^{[t]}\right\|^2\right] = \underbrace{\left\|\mathbb{E}[\widehat{\boldsymbol{d}}^{[t]}] - \boldsymbol{d}^{[t]}\right\|^2}_{\text{bias}^2} + \underbrace{\mathbb{E}\left[\left\|\widehat{\boldsymbol{d}}^{[t]} - \mathbb{E}[\widehat{\boldsymbol{d}}^{[t]}]\right\|^2\right]}_{\text{variance}}, \tag{2.69}$$

is sufficiently small such that the optimizer can make meaningful progress. However, note that although we are interested in low MSE gradient estimates, a small amount of noise can be beneficial for the optimization process as it can help to escape local minima and so find more stable solutions. The bias-variance trade-off illustrated by the MSE is fundamental: on one hand, we want an unbiased estimator with low variance; on the other hand, even perfectly unbiased estimators may perform poorly if their variance is high, and a small bias leading to reduced variance can yield a substantial improvement in the optimization process.

Interchanging Gradients and Expectations. In variational inference, we often have $\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{z}\sim Q_{\boldsymbol{\theta}}}[f_{\boldsymbol{\theta}}(\boldsymbol{z})]$ for some $f_{\boldsymbol{\theta}}$ depending on both $\boldsymbol{z}$ and $\boldsymbol{\theta}$. Typically, there is no closed-form expression for the gradient $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})$, which is why we rely on sample-based (Monte Carlo) solutions. The difficulty arises because the expectation itself depends on $\boldsymbol{\theta}$: we cannot simply interchange the gradient and the expectation without introducing bias.

[8] We overload notation and write $\mathcal{L}(\boldsymbol{\theta})$ to indicate dependence of $\mathcal{L}$ on $\boldsymbol{\theta}$ through $Q_{\boldsymbol{\theta}}$.

The score-function (REINFORCE) estimator (Williams, 1992) was the first unbiased gradient estimator proposed for this problem. Under standard regularity conditions, the score-function estimator is unbiased and given by

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim Q_{\boldsymbol{\theta}}}[f_{\boldsymbol{\theta}}(\boldsymbol{z})] = \nabla_{\boldsymbol{\theta}} \int_{\mathcal{Z}} q_{\boldsymbol{\theta}}(\boldsymbol{z}) f_{\boldsymbol{\theta}}(\boldsymbol{z}) d\boldsymbol{z} = \mathbb{E}_{\boldsymbol{z}\sim Q_{\boldsymbol{\theta}}}\left[f_{\boldsymbol{\theta}}(\boldsymbol{z}) \nabla_{\boldsymbol{\theta}} \log q_{\boldsymbol{\theta}}(\boldsymbol{z}) + \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{z})\right] \tag{2.70}$$

which follows because $\nabla_{\boldsymbol{\theta}} q_{\boldsymbol{\theta}}(\boldsymbol{z}) = q_{\boldsymbol{\theta}}(\boldsymbol{z}) \nabla_{\boldsymbol{\theta}} \log q_{\boldsymbol{\theta}}(\boldsymbol{z})$ and the chain rule. While the score-function estimator is unbiased, it typically suffers from high variance and usually requires further variance reduction techniques to be practical (Ranganath et al., 2014).

When applicable, the reparameterization gradient estimator (Kingma and Welling, 2014; Rezende et al., 2014) has, in practice, replaced the score-function estimator in modern VI due to its often substantial reduction in variance while still providing unbiased gradient estimates. The reparameterization gradient estimator is valid under the following assumptions:

- $Q_{\boldsymbol{\theta}}$ is reparameterizable; that is, $\boldsymbol{z} = T_{\boldsymbol{\theta}}(\boldsymbol{\epsilon})$, where $\boldsymbol{\epsilon} \sim \pi$, $T_{\boldsymbol{\theta}}$ is a differentiable function of $\boldsymbol{\theta}$ (for almost every $\boldsymbol{\epsilon}$), and π is a distribution independent of $\boldsymbol{\theta}$,
- $f_{\boldsymbol{\theta}}(\boldsymbol{z})$ is differentiable in $\boldsymbol{\theta}$ for (almost) every $\boldsymbol{z}$,
- there exists an integrable function $g(\boldsymbol{\epsilon})$ such that $\|\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(T_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}))\| \leq g(\boldsymbol{\epsilon})$ and $\mathbb{E}_{\boldsymbol{\epsilon}\sim\pi}[g(\boldsymbol{\epsilon})] < \infty$.

Under these conditions, the reparameterization gradient is given by

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim Q_{\boldsymbol{\theta}}}[f_{\boldsymbol{\theta}}(\boldsymbol{z})] = \mathbb{E}_{\boldsymbol{\epsilon}\sim\pi}\left[\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}\left(T_{\boldsymbol{\theta}}(\boldsymbol{\epsilon})\right)\right]. \tag{2.71}$$

In contrast to the score-function estimator, which relies only on the function values $f_{\boldsymbol{\theta}}(\boldsymbol{z})$, the reparameterization gradient requires $f_{\boldsymbol{\theta}}(\boldsymbol{z})$ to be differentiable with respect to $\boldsymbol{z}$ almost everywhere. However, this requirement is also one of the main reasons for its practical success: whereas the score-function estimator ignores the internal structure of $f_{\boldsymbol{\theta}}(\boldsymbol{z})$, the reparameterization gradient can exploit it through the chain rule, yielding a more informative and lower-variance gradient.

The reparameterization gradient estimator can also be applied when the inverse transformation $T_{\boldsymbol{\theta}}^{-1}$ is available, even if the sampling process itself is not directly reparameterizable. In such cases, the implicit reparameterization trick (Figurnov et al., 2018), which leverages the inverse function theorem and the total derivative, can be used to compute gradients.

When $Q_{\boldsymbol{\theta}}$ has a differentiable density $q_{\boldsymbol{\theta}}(\boldsymbol{z})$ and is reparameterizable ($\boldsymbol{z} = T_{\boldsymbol{\theta}}(\boldsymbol{\epsilon})$, $\boldsymbol{\epsilon} \sim \pi$), the expectation $\mathbb{E}_{\boldsymbol{z}\sim Q_{\boldsymbol{\theta}}}[\log q_{\boldsymbol{\theta}}(\boldsymbol{z})]$ admits a *pathwise* (reparameterization) gradient estimator (Roeder et al., 2017):

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim Q_{\boldsymbol{\theta}}}[\log q_{\boldsymbol{\theta}}(\boldsymbol{z})] = \mathbb{E}_{\boldsymbol{\epsilon}\sim\pi}\left[\nabla_{\boldsymbol{z}} \log q_{\boldsymbol{\theta}}(\boldsymbol{z})\Big|_{\boldsymbol{z}=T_{\boldsymbol{\theta}}(\boldsymbol{\epsilon})} \cdot \nabla_{\boldsymbol{\theta}} T_{\boldsymbol{\theta}}(\boldsymbol{\epsilon})\right]. \tag{2.72}$$

This expression arises by taking the total derivative with respect to $\boldsymbol{\theta}$ and noting that the expected score term $\mathbb{E}_{\boldsymbol{z}\sim Q_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \log q_{\boldsymbol{\theta}}(\boldsymbol{z})\right]$ is zero. Importantly, in contrast to the standard reparameterization trick, we only need to differentiate $q_{\boldsymbol{\theta}}(\boldsymbol{z})$ with respect to $\boldsymbol{z}$, not directly with respect to $\boldsymbol{\theta}$. All contributions of this work use the path gradient, since the score gradient $\nabla_{\boldsymbol{z}} \log q_{\boldsymbol{\theta}}(\boldsymbol{z})$ can be efficiently estimated via sampling even when the density $q_{\boldsymbol{\theta}}(\boldsymbol{z})$ is not available in closed form.

KL and Score-Based Objectives in Variational Inference. In variational inference, the target density is, in general, unnormalized. Often this occurs when the target distribution is a conditional distribution with $p(\boldsymbol{z} \mid \boldsymbol{u})$, but we only have direct access to the joint density $p(\boldsymbol{z}, \boldsymbol{u}) = p(\boldsymbol{z})p(\boldsymbol{u} \mid \boldsymbol{z})$, while the marginal likelihood $p(\boldsymbol{u})$ is intractable as in Bayesian inference, where $\boldsymbol{u}$ is the observable (see Section 2.3). KL-based methods avoid this difficulty because of their normalization invariance from an optimization perspective, i.e., $\mathrm{D}_{\mathrm{KL}}(q_{\boldsymbol{\theta}} \| p/Z) = \mathrm{D}_{\mathrm{KL}}(q_{\boldsymbol{\theta}} \| p) + \log Z$, and specifically for a conditional target distribution,

$$\mathrm{D}_{\mathrm{KL}}(q_{\boldsymbol{\theta}} \| p(\cdot \mid \boldsymbol{u})) = \underbrace{\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\theta}}}[\log q_{\boldsymbol{\theta}}(\boldsymbol{z})] - \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\theta}}}[\log p(\boldsymbol{z}, \boldsymbol{u})]}_{=: -\mathrm{ELBO}(\boldsymbol{\theta})} + \log p(\boldsymbol{u}). \tag{2.73}$$

Hence, maximizing the *evidence lower bound* (ELBO) is equivalent to minimizing the KL up to an irrelevant additive constant, which in the conditional case is the log of the marginal likelihood (evidence) $\log p(\boldsymbol{u})$. Score-based methods rely on the same normalization invariance, since

$$\nabla_{\boldsymbol{z}} \log(p/Z) = \nabla_{\boldsymbol{z}} \log p \quad \text{and consequently} \quad \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z} \mid \boldsymbol{u}) = \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}, \boldsymbol{u}), \tag{2.74}$$

and therefore never require the evidence either. When the likelihood factorizes over the observable, $p(\boldsymbol{u} \mid \boldsymbol{z}) = \prod_{i=1}^{N} p(\boldsymbol{u}^{(i)} \mid \boldsymbol{z})$, both ELBO/KL gradients and score-based objectives admit unbiased minibatch estimators by replacing the full sum $\sum_{i=1}^{N} \log p(\boldsymbol{u}^{(i)} \mid \boldsymbol{z})$ (or its gradient) with a minibatch average of size B scaled by N/B. These properties make KL/ELBO and score-based objectives well suited to high-dimensional inference and large datasets, since they admit efficient minibatch-based gradient estimates and remain invariant to unknown normalizing constants.

2.5.2. Explicit Variational Inference

We refer to explicit variational inference as the classical case in which the variational family is chosen so that the density function $q_{\boldsymbol{\theta}}$ is analytically available and differentiable; that is, for any $(\boldsymbol{z}, \boldsymbol{\theta})$, $q_{\boldsymbol{\theta}}(\boldsymbol{z})$ can be evaluated directly, not only sampled from. This property is essential for applying standard VI objectives and gradient-based optimization, as it enables direct evaluation and differentiation of (log-)densities, the use of score-function estimators, and unbiased estimation of quantities such as entropy or KL divergence.

In this chapter, we assume that the variational families under consideration are reparameterizable, i.e., $\boldsymbol{z} = T_{\boldsymbol{\theta}}(\boldsymbol{\epsilon})$, where $T_{\boldsymbol{\theta}}$ is a differentiable function of $\boldsymbol{\theta}$ and $\boldsymbol{\epsilon} \sim \pi$ is a distribution independent of $\boldsymbol{\theta}$. This includes simple mean-field families, such as factorized Gaussian families where $T_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}) = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$ with $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\sigma}) \in \mathbb{R}^d \times \mathbb{R}^d_+$ and $\pi = \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, as well as more expressive families such as normalizing flows, which will be discussed later in Section 2.5.2.

Variational Inference via Reverse KL. For explicit variational inference, the standard objective is the *reverse* KL divergence between the variational distribution $Q_{\boldsymbol{\theta}}$ and the target distribution P given by a (possibly unnormalized) density $p(\boldsymbol{z})$,

$$\mathrm{KL}(Q_{\boldsymbol{\theta}} \,\|\, P) = \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\theta}}}\left[\log \frac{q_{\boldsymbol{\theta}}(\boldsymbol{z})}{p(\boldsymbol{z})}\right], \tag{2.75}$$

because of the beneficial properties already discussed (see Sections 2.4.4 and 2.5.1) and because it is analytically tractable by assumption. With the reparameterization $\boldsymbol{z} = T_{\boldsymbol{\theta}}(\boldsymbol{\epsilon})$ where $\boldsymbol{\epsilon} \sim \pi$, the gradient of the reverse KL objective can be written as

$$\nabla_{\boldsymbol{\theta}} \mathrm{KL}(Q_{\boldsymbol{\theta}} \,\|\, P) = \mathbb{E}_{\boldsymbol{\epsilon}\sim\pi} \left[\nabla_{\boldsymbol{\theta}} \log q_{\boldsymbol{\theta}}(T_{\boldsymbol{\theta}}(\boldsymbol{\epsilon})) - \nabla_{\boldsymbol{\theta}} \log p(T_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}))\right] \tag{2.76}$$

which we can straightforwardly estimate via Monte Carlo.

Normalizing Flows. Among the variational inference families with explicit and tractable densities, normalizing flows (Rezende and Mohamed, 2015; Papamakarios et al., 2021) are one of the most expressive approaches. They transform a simple base distribution π into a more complex one using a sequence of invertible, differentiable mappings. Given $\boldsymbol{z}_0 = \boldsymbol{\epsilon} \sim \pi$ and successive transformations $T_{\boldsymbol{\theta},1}, \ldots, T_{\boldsymbol{\theta},K}$ (with $\boldsymbol{z}_k = T_{\boldsymbol{\theta},k}(\boldsymbol{z}_{k-1})$), the overall mapping is $T_{\boldsymbol{\theta}} := T_{\boldsymbol{\theta},K} \circ \cdots \circ T_{\boldsymbol{\theta},1}$, so $\boldsymbol{z}_K = T_{\boldsymbol{\theta}}(\boldsymbol{z}_0)$. The density of the final variable $\boldsymbol{z}_K$ follows from the change of variables formula:

$$q_{\boldsymbol{\theta}}(\boldsymbol{z}_K) = q_\pi(\boldsymbol{z}_0) \left|\det \frac{\partial \boldsymbol{z}_K}{\partial \boldsymbol{z}_0}\right|^{-1} = q_\pi(\boldsymbol{z}_0) \prod_{k=1}^{K} \left|\det \frac{\partial T_{\boldsymbol{\theta},k}(\boldsymbol{z}_{k-1})}{\partial \boldsymbol{z}_{k-1}}\right|^{-1}, \tag{2.77}$$

where q_π is the density of the base distribution π and $q_{\boldsymbol{\theta}}$ is the density of the final variable $\boldsymbol{z}_K$. This allows both sampling and exact density evaluation as long as each $T_{\boldsymbol{\theta},k}$ and its Jacobian determinant are tractable to compute. By chaining tractable, expressive mappings, flows model distributions of high complexity while keeping explicit densities accessible.

A key design challenge is ensuring efficient computation of Jacobian determinants, as this underlies likelihood evaluation. Architectural innovations such as affine coupling layers (Dinh et al., 2017; Kingma and Dhariwal, 2018), neural autoregressive flows (Huang et al., 2018), and neural spline flows (Durkan et al., 2019) have been developed to make both sampling and determinant calculations practical, balancing expressiveness with computational efficiency.

Nevertheless, the requirement that transformations need to be invertible and differentiable imposes inductive bias: only smooth, non-singular mappings are permitted, so extremely sharp or discontinuous distributions cannot be represented. As a result, flows tend toward smooth densities, and their architecture must avoid regions where the Jacobian is nearly singular, which affects both flexibility and numerical stability (Papamakarios et al., 2021). Also, in practice, flows can model heavy tails only if the base distribution is heavy-tailed (Laszkiewicz et al., 2022), since standard flow transformations do not exhibit the asymptotic growth needed to induce heavy tails on their own. Therefore, although some flows are universal approximators in theory, practical flows may still struggle to represent complex, multimodal, or sharply peaked, heavy-tailed target distributions, especially as dimension increases.

2.5.3. Particle-Based Variational Inference

In particle-based variational inference (PVI), the variational distribution Q is represented by an empirical measure supported on a set of particles $\{\boldsymbol{z}^{(i)}\}_{i=1}^{N}$. Instead of modeling Q with a specific generative process, we maintain a collection of particles and update their positions to approximate the target distribution P. The particles are typically updated such that

$$\boldsymbol{z}^{(i)[t+1]} \leftarrow \boldsymbol{z}^{(i)[t]} + \eta^{[t]}\, \boldsymbol{\phi}\left(\boldsymbol{z}^{(i)[t]}\right), \tag{2.78}$$

where $\boldsymbol{\phi}$ is a velocity field and $\eta^{[t]}$ is the learning rate at iteration t. This framework is highly flexible: with enough particles and appropriate updates, PVI can, in principle, approximate any distribution P.

Different PVI algorithms can be distinguished by the divergence they implicitly minimize. When the functional gradient of the KL divergence is restricted to an RKHS, the resulting method is Stein variational gradient descent (SVGD; Liu and Wang, 2016). The flexibility of such kernel-based frameworks introduces both computational and statistical trade-offs that depend on the divergence, kernel, and number of particles. For instance, SVGD and related algorithms tend to have quadratic complexity in the number of particles and their performance is sensitive to the kernel choice, often limiting practical scalability to high-dimensional problems (Liu and Wang, 2016; Korba et al., 2021).

2.5.4. Implicit Variational Inference

Implicit variational inference specifies the variational family only through a sampling mechanism $\boldsymbol{z} = T_{\boldsymbol{\theta}}(\boldsymbol{\epsilon})$, with $\boldsymbol{\epsilon} \sim \pi$, without requiring $T_{\boldsymbol{\theta}}$ to be invertible (Huszár, 2017; Tran et al., 2017). This construction allows highly flexible neural-network-based parameterizations and can represent complex or multimodal target distributions, but its variational density $q_{\boldsymbol{\theta}}(\boldsymbol{z})$ is, in general, intractable. Hence, direct application of standard KL-based optimization is not possible. To address this difficulty, three broad classes of methods have been developed:

- **Discriminator-based methods:** These introduce a classifier trained to distinguish samples from the variational distribution and a reference distribution, yielding an *implicit* estimate of the density ratio $p(\boldsymbol{z})/q_{\boldsymbol{\theta}}(\boldsymbol{z})$ through adversarial optimization (Tran et al., 2017; Mescheder et al., 2017).
- **Direct density-ratio estimation methods:** These approximate the density ratio explicitly via regression or moment matching, avoiding an adversarial setup but inheriting the statistical challenges of ratio estimation, particularly in high dimensions (Sugiyama et al., 2012; Huszár, 2017; Shi et al., 2018).
- **Stein-based methods:** These leverage Stein identities to construct gradient estimators that depend only on the target score $\nabla_{\boldsymbol{z}} \log p(\boldsymbol{z})$ and do not require evaluating the variational density. Typically, kernel or score-matching constructions are used to define practical algorithms (Feng et al., 2017; Li and Turner, 2018).

Overall, implicit VI substantially broadens the space of admissible variational families, but the intractability of $q_{\boldsymbol{\theta}}(\boldsymbol{z})$ necessitates auxiliary mechanisms, each providing a different route to workable gradients with distinct methodological trade-offs.

2.5.5. Semi-Implicit Variational Inference

For Semi-Implicit Variational Inference (SIVI) (Yin and Zhou, 2018), we define the variational family

$$\mathcal{Q} = \left\{ Q_{\boldsymbol{\theta}} \text{ with density } q_{\boldsymbol{\theta}} \;\middle|\; q_{\boldsymbol{\theta}}(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{\epsilon}\sim p_{\boldsymbol{\epsilon}}}\left[q_{\boldsymbol{z}|\boldsymbol{\psi}}(\boldsymbol{z} \mid \boldsymbol{\psi} = f_{\boldsymbol{\theta}}(\boldsymbol{\epsilon})) \right], \boldsymbol{\theta} \in \Theta \right\}, \tag{2.79}$$

where $p_{\boldsymbol{\epsilon}}$ is the density of the base distribution and $q_{\boldsymbol{z}|\boldsymbol{\psi}}$ is a tractable density with parameters $\boldsymbol{\psi}$, for example a parameterized Gaussian. The function $f_{\boldsymbol{\theta}}$ is a differentiable mapping (typically a neural network) from the noise variable $\boldsymbol{\epsilon}$ to the parameter vector $\boldsymbol{\psi}$. Hence, to generate samples from $q_{\boldsymbol{\theta}}(\boldsymbol{z})$, we first sample $\boldsymbol{\epsilon} \sim p_{\boldsymbol{\epsilon}}$, and then draw $\boldsymbol{z}$ from $q_{\boldsymbol{z}|\boldsymbol{\psi}}(\cdot \mid \boldsymbol{\psi} = f_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}))$.

SIVI inherits its flexibility from the implicit mixing distribution over $\boldsymbol{\psi}$ while at the same time providing a more structured variational family. In principle, a semi-implicit model can be treated as a fully implicit variational distribution, allowing one to apply the techniques discussed in Section 2.5.4. However, more robust and efficient training methods have been developed specifically for SIVI, which explicitly exploit its semi-implicit structure. We discuss these SIVI approaches and further details in Chapter III.

2.5.6. Functional Variational Inference

Functional variational inference (FVI) takes a more general view by treating VI as optimization directly over stochastic processes with respect to a suitable divergence. In this work, we focus on Bayesian inference in the supervised setting in which the target P is a posterior stochastic process (see Section 2.3), which defines a distribution over functions mapping from the input space $\mathcal{X}$ to the output space $\mathcal{Y}$ where we assume that both $\mathcal{X}$ and $\mathcal{Y}$ are real-valued, continuous vector spaces. Modern FVI approaches model the variational stochastic process Q implicitly through its sampling mechanism: for $\boldsymbol{z} \sim Q$, we write $\boldsymbol{z} = T_{\boldsymbol{\theta}}(\cdot, \boldsymbol{\epsilon})$ with $\boldsymbol{\epsilon} \sim \pi(\boldsymbol{\epsilon})$, where $T_{\boldsymbol{\theta}} : \mathcal{X} \times \mathcal{E} \to \mathcal{Y}$ is a differentiable mapping (typically a neural network) and π is a base distribution on a finite-dimensional real space $\mathcal{E}$. In contrast to Bayesian neural networks, where priors are placed over the parameters of an overparameterized neural network, FVI places priors directly over functions, which can lead to more interpretable and meaningful expressions of prior knowledge. However, as discussed in Section 2.3, the consistency of FVI cannot be guaranteed in general. See Chapter II for more details and discussion of specific FVI approaches.

2.5.7. Summary and Comparison of VI Approaches

This section introduced five major paradigms for variational inference, each representing distinct trade-offs between tractability, expressivity, and computational efficiency. Explicit VI (Section 2.5.2) provides tractable densities and reliable gradient estimates but is generally limited in expressivity, particularly for multimodal or heavy-tailed targets. Implicit VI (Section 2.5.4) removes most architectural constraints to achieve greater flexibility, though at the cost of requiring auxiliary training mechanisms such as discriminators or density-ratio estimators. Semi-implicit VI (Section 2.5.5) strikes a middle ground by maintaining tractable conditionals while marginalizing over an implicit distribution, enabling efficient gradient estimation via the hierarchical structure. Particle-based VI (Section 2.5.3) avoids parametric assumptions entirely but typically faces quadratic computational costs. Functional VI (Section 2.5.6) operates directly in function space, naturally allowing for meaningful priors over functions, but consistency cannot be guaranteed. Table 2.2 summarizes the key distinguishing properties of these approaches, highlighting that modern VI increasingly favors generator-based constructions that balance expressivity with computational tractability.

Property	Explicit	Implicit	Semi-Implicit	Particle	Functional
q available	Yes	No	$q(\boldsymbol{z} \mid \boldsymbol{\psi})$ only	No	—
Generator-based	Yes	Yes	Yes	No	Yes
Expressivity	Medium	High	High	Very High*	High
Training tractability	Simple	Complex	Moderate	Moderate	Complex
Scalability	High	Medium	Medium-High	Low-Medium	Medium

* Expressivity is maximal asymptotically, as empirical measures can approximate any target distribution (including disconnected modes).

Table 2.2.: Comparison of the main variational inference approaches and their properties.

2.6. Evaluation and Tuning of VI Methods

The *evaluation* of variational inference methods is crucial for understanding how successfully the approximation captures the target distribution, and for diagnosing both failure modes and areas for improvement. Since these evaluation outcomes are usually strongly influenced by some of the hyperparameters of the VI algorithm, *tuning* tries to find the optimal hyperparameters for a given VI algorithm. The strategy for evaluation and tuning depends fundamentally on the inference scenario: for example, tractable generative settings allow direct quantitative measures, while Bayesian learning with intractable posteriors often requires predictive assessments.

In general, for monitoring the convergence of the VI algorithm, we can use, for example, the VI loss $\mathcal{L}$, its gradient norm $\|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})\|$, and the parameter update norms $\|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|$. However, since typical VI losses $\mathcal{L}$ are estimated using samples from the variational distribution Q, all terms in the objective are effectively weighted by Q rather than by the target P. This makes the objective inherently mode-seeking and allows convergence to local optima in which Q captures only a subset of the target probability mass (see Section 2.4). Hence, the loss $\mathcal{L}$ is usually not a meaningful indicator of the approximation quality of Q. When q is tractable, we can also examine the ESS when using q as a proposal (see Section 2.1). This diagnostic can reveal severe mismatch within the support of Q, but it cannot detect mismatch in regions Q assigns negligible mass to.

2.6.1. Assessing Approximation Quality for Intractable Targets

Here the goal is to assess how well the variational distribution q approximates the (usually unnormalized) target density p. In this setting, meaningful evaluation metrics can only be obtained from (approximate) samples of p. When genuine samples from p are available, such as in synthetic settings or in physical scenarios where we have access to a simulator capable of generating samples from the target (often computationally intensive, e.g., molecular dynamics), we prefer to use them for evaluation since they provide an unbiased reference. Otherwise, we often rely on approximate samples from Markov chain Monte Carlo (MCMC) methods. While MCMC methods have strong theoretical guarantees, they can be computationally expensive and may still struggle in practice, especially in high-dimensional settings, but are often the only practical choice when the target is intractable. In both cases, the same general evaluation techniques can be applied, for example:

- **Sample-based divergence estimates.** Estimate discrepancies such as the Wasserstein distance (and its variants) or MMD between samples from q and p (see Section 2.4.3). These metrics are general but can be computationally expensive. Both Wasserstein distances and

MMD control only weak convergence, although Wasserstein distances additionally control moments up to a certain order (see Section 2.4).

- **Cross-entropy evaluation.** If the variational log density $\log q(\boldsymbol{z})$ is tractable (or can be efficiently estimated), estimate the cross-entropy $-\mathbb{E}_{\boldsymbol{z}\sim p}[\log q(\boldsymbol{z})]$ using samples from p. Unlike sample-based divergences, the cross-entropy does not yield a canonical value corresponding to a perfect match between q and p. This is because cross-entropy differs from the KL divergence by the entropy $H(p) = -\mathbb{E}_{\boldsymbol{z}\sim p}[\log p(\boldsymbol{z})]$ of the target:

$$-\mathbb{E}_{\boldsymbol{z}\sim p}[\log q(\boldsymbol{z})] = \mathrm{D}_{\mathrm{KL}}(p \,\|\, q) + H(p). \tag{2.80}$$

  Since the entropy of p is rarely available except in synthetic cases, the KL divergence cannot be estimated. Nevertheless, cross-entropy remains useful for comparing the relative quality of different approximations, since $H(p)$ does not depend on the variational distribution q.

- **Comparison of key summary statistics.** Compare means, variances, or other moments of key quantities under q and p.

- **Visualizations.** Plot marginals or low-dimensional projections of q and p to visually assess approximation quality.

In this setting, hyperparameter tuning may involve using (approximate) target samples for internal evaluations. In this case, it is crucial to ensure that samples used for hyperparameter selection are not reused for final evaluation, as this can lead to information leakage and biased approximation-quality estimates. Similarly, different random seeds should be used during tuning and final evaluation to reduce the risk of overfitting to particular random draws.

2.6.2. Assessing Predictive Performance in Conditional Models

In supervised learning with Bayesian inference (see Section 2.3), if the latent variables $\boldsymbol{z}$ are low-dimensional and interpretable (such as in linear regression), the parameter posterior distribution itself is a meaningful object of study. In these cases, we should ideally use direct approximation quality evaluation methods as described in the previous Section 2.6.1. Conversely, as we move to more complex or overparameterized models such as Bayesian neural networks where individual parameters (like neural network weights) become less meaningful, or settings involving functional variational inference, the posterior distribution over $\boldsymbol{z}$ becomes less interpretable. Simultaneously, direct comparison between the approximate and true posterior becomes impractical or infeasible. Therefore, the evaluation focus naturally shifts away from the posterior distribution itself toward the *predictive* perspective, where the primary object of interest is the posterior predictive distribution. As explained in Section 2.3, in Bayesian supervised learning, for realized observable $\widetilde{\boldsymbol{u}} = \left(\widetilde{\boldsymbol{y}}^{(1)}, \ldots, \widetilde{\boldsymbol{y}}^{(N)}\right)$ conditioning on the inputs $\widetilde{\boldsymbol{X}} = \left(\widetilde{\boldsymbol{x}}^{(1)}, \ldots, \widetilde{\boldsymbol{x}}^{(N)}\right)$ the density of the posterior predictive distribution for new observations $\boldsymbol{y}^{*(1)}, \ldots, \boldsymbol{y}^{*(M)}$ given inputs $\boldsymbol{X}^* = \left(\boldsymbol{x}^{*(1)}, \ldots, \boldsymbol{x}^{*(M)}\right)$ is given by

$$p\left(\boldsymbol{y}^{*(1)}, \ldots, \boldsymbol{y}^{*(M)} \mid \boldsymbol{X}^*, \widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{X}}\right) = \int_{\mathcal{Z}} \prod_{i=1}^{M} l\left(\boldsymbol{y}^{*(i)} \mid \boldsymbol{x}^{*(i)}, \boldsymbol{z}\right) \Pi(d\boldsymbol{z} \mid \widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{X}}), \tag{2.81}$$

where $l\left(\boldsymbol{y}^{*(i)} \mid \boldsymbol{x}^{*(i)}, \boldsymbol{z}\right)$ is the likelihood for output $\boldsymbol{y}^{*(i)}$ given input $\boldsymbol{x}^{*(i)}$ and $\boldsymbol{z}$. With a variational approximation $Q(\boldsymbol{z})$ of the posterior, this approximate predictive density becomes

$$\widehat{p}\left(\boldsymbol{y}^{*(1)}, \ldots, \boldsymbol{y}^{*(M)} \mid \boldsymbol{X}^*, \widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{X}}\right) = \int_{\mathcal{Z}} \prod_{i=1}^{M} l(\boldsymbol{y}^{*(i)} \mid \boldsymbol{x}^{*(i)}, \boldsymbol{z})\, Q(d\boldsymbol{z}). \tag{2.82}$$

Note that although outputs are conditionally independent given $\boldsymbol{z}$, posterior predictive samples are not independent: dependence is induced by integrating out the shared random variable $\boldsymbol{z}$. Hence predictions for a set of inputs must be sampled jointly, i.e., we first draw $\boldsymbol{z} \sim Q$ and then draw $\boldsymbol{y}^{*(i)} \sim l\left(\cdot \mid \boldsymbol{x}^{*(i)}, \boldsymbol{z}\right)$. So in this setting, the VI algorithm $\mathcal{I}_{\boldsymbol{\lambda}}$ maps a *training data set* $\mathcal{D}_{\text{train}} = \left(\left(\widetilde{\boldsymbol{x}}^{(1)}, \widetilde{\boldsymbol{y}}^{(1)}\right) \ldots, \left(\widetilde{\boldsymbol{x}}^{(N)}, \widetilde{\boldsymbol{y}}^{(N)}\right)\right) \sim P_{\boldsymbol{xy}}^{\otimes N}$ to a probabilistic model $\widehat{h}$. A probabilistic model $\widehat{h}$ in turn, maps a set of inputs $\boldsymbol{X}^* \in \cup_{M \in \mathbb{N}} \mathcal{X}^M$ to the (approximate) predictive distribution $\widehat{p}(\cdot \mid \boldsymbol{X}^*, \widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{X}})$. This setting closely parallels the machine learning paradigm, where a ML learner maps a training data set $\mathcal{D}_{\text{train}}$ to a model $\widehat{f}$ that subsequently takes an input $\boldsymbol{x}^* \in \mathcal{X}$ and produces a (possibly encoded) prediction $\widehat{f}(\boldsymbol{x}^*)$ (Bischl et al., 2023). This observation suggests an evaluation framework similar to the classical machine learning setting based on a second data set independent of the training set to evaluate the predictive performance of the model, i.e., a test set $\mathcal{D}_{\text{test}} = \left(\left(\boldsymbol{x}^{*(1)}, \boldsymbol{y}^{*(1)}\right) \ldots, \left(\boldsymbol{x}^{*(M)}, \boldsymbol{y}^{*(M)}\right)\right) \sim P_{\boldsymbol{xy}}^{\otimes M}$, but adapted to distribution-valued predictions:

- **Test negative log likelihood.** A probabilistic learner $\widehat{h}$ is equipped with a natural evaluation metric, the negative log likelihood (NLL). The test NLL is given by

$$\text{NLL}_{\text{test}} = -\log \widehat{h}(\boldsymbol{X}^*)(\boldsymbol{y}^{*(1)}, \ldots, \boldsymbol{y}^{*(M)}) = -\log \int_{\mathcal{Z}} \prod_{i=1}^{M} l(\boldsymbol{y}^{*(i)} \mid \boldsymbol{x}^{*(i)}, \boldsymbol{z})\, Q(d\boldsymbol{z}). \tag{2.83}$$

  The Monte Carlo estimate of the test NLL is given by

$$\widehat{\text{NLL}}_{\text{test}} = -\log\left(\frac{1}{S}\sum_{s=1}^{S}\prod_{i=1}^{M} l\left(\boldsymbol{y}^{*(i)} \mid \boldsymbol{x}^{*(i)}, \boldsymbol{z}^{(s)}\right)\right), \tag{2.84}$$

  where $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(S)} \overset{\text{i.i.d.}}{\sim} Q$. By the law of large numbers, $\widehat{\text{NLL}}_{\text{test}}$ converges almost surely to NLL_{test} as $S \to \infty$ but by Jensen's inequality it is a biased estimator. So in order to get reliable estimates, we need to use a sufficiently large number of samples S. In practice, we rewrite the estimate as

$$\widehat{\text{NLL}}_{\text{test}} = -\log\left(\sum_{s=1}^{S}\exp\left(\sum_{i=1}^{M}\log l\left(\boldsymbol{y}^{*(i)} \mid \boldsymbol{x}^{*(i)}, \boldsymbol{z}^{(s)}\right)\right)\right) + \log S. \tag{2.85}$$

  which can be computed in a numerically stable way using the log-sum-exp trick (Murphy, 2012).

- **Pointwise prediction losses.** In many practical applications, one evaluates summaries of the predictive distribution such as the *predictive mean*

$$\widehat{\boldsymbol{y}}^{*(i)} = \mathbb{E}_{\boldsymbol{y}^{*(i)} \sim \widehat{h}(\boldsymbol{x}^{*(i)})}\left[\boldsymbol{y}^{*(i)}\right], \tag{2.86}$$

  and then applies standard ML evaluation metrics to these point predictions. Common examples include in regression settings the root mean squared error (RMSE), the mean

absolute error (MAE), or, in classification settings, accuracy or cross-entropy computed from the predictive mean or mode. Since these metrics rely only on summary statistics of the posterior predictive distribution, they provide complementary diagnostics to the NLL, which evaluates the full predictive distribution.

- **Visualization of predictive performance.** Qualitative diagnostics are often useful to detect systematic deviations not captured by scalar metrics. Typical visual tools include for example plots of predictive means with credible intervals against observed test outputs and overlays of posterior predictive densities with empirical test-data distributions. These visualizations provide interpretable insights into general model fit.

In supervised learning, the close analogy between VI algorithms and standard ML learners naturally extends to hyperparameter tuning. In practice, one may tune hyperparameters $\boldsymbol{\lambda}$ of the VI algorithm $\mathcal{I}_{\boldsymbol{\lambda}}$ using standard ML resampling strategies such as cross-validation in an inner resampling loop, and assess the resulting model's generalization performance using nested resampling in the outer loop, as described by Bischl et al. (2023). When tuning with respect to the test NLL, it is important to note that the optimization target is not matching the posterior distribution over latent variables, but purely the quality of the posterior predictive distribution. A lower NLL indicates that the approximate predictive distribution with density $\widehat{p}(\cdot \mid \boldsymbol{x}^*, \widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{X}})$ is closer to the true data-generating distribution $P_*(\cdot \mid \boldsymbol{x}^*)$, but it does not imply a better approximation of the true Bayesian posterior $\Pi(\mathrm{d}\boldsymbol{z} \mid \widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{X}})$. Moreover, a model achieving strong predictive NLL does not necessarily yield optimal pointwise predictive performance (e.g., as measured by RMSE or MAE). In principle, one could tune hyperparameters jointly with respect to multiple evaluation criteria, such as NLL and a pointwise predictive metric, leading to a multi-objective hyperparameter optimization problem (Karl et al., 2023). However, such multi-objective tuning strategies are beyond the scope of this thesis.

Part II.

Contribution to Functional Variational Inference

3. Approximate Bayesian Inference with Stein Functional Variational Gradient Descent

Contributing article

Tobias Pielok, Bernd Bischl, and David Rügamer. 2023. Approximate Bayesian Inference with Stein Functional Variational Gradient Descent. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/pdf?id=a2-aoqmeYM4.

Copyright information

Copyright © 2023 by the authors and ICLR.

Author contributions

Tobias Pielok developed the core ideas, the full theoretical framework, and all implementations, except for the Contextual Bandits benchmark, which was conducted by David Rügamer. Tobias Pielok also independently carried out all conceptual contributions, mathematical derivations, and remaining experiments. David Rügamer contributed through close collaboration on the structure, formulation, and overall readability of the manuscript. Bernd Bischl provided valuable input on the presentation and emphasis of the work, as well as detailed feedback on writing and benchmarking strategy.

Slides https://slideslive.com/38998794/approximate-bayesian-inference-with-stein-functional-variational-gradient-descent

Published as a conference paper at ICLR 2023

APPROXIMATE BAYESIAN INFERENCE WITH STEIN FUNCTIONAL VARIATIONAL GRADIENT DESCENT

Tobias Pielok, Bernd Bischl, David Rügamer
Department of Statistics, LMU Munich, Munich, Germany
Munich Center for Machine Learning, Munich, Germany
{tobias.pielok, bernd.bischl, david.ruegamer}@stat.uni-muenchen.de

ABSTRACT

We propose a general-purpose variational algorithm that forms a natural analogue of Stein variational gradient descent (SVGD) in function space. While SVGD successively updates a set of particles to match a target density, the method introduced here of Stein functional variational gradient descent (SFVGD) updates a set of particle functions to match a target stochastic process (SP). The update step is found by minimizing the functional derivative of the Kullback-Leibler divergence between SPs. SFVGD can either be used to train Bayesian neural networks (BNNs) or for ensemble gradient boosting. We show the efficacy of training BNNs with SFVGD on various real-world datasets.

1 INTRODUCTION

Bayesian inference can be treated as a powerful framework for data modeling and reasoning under uncertainty. However, this assumes that we can encode our prior knowledge in a meaningful manner. Typically, this is done by specifying the prior distribution of the model parameters. However, in machine learning (ML), models potentially consist of millions of parameters with potentially highly complex interactions (e.g., very large neural networks (NNs)). Furthermore, the parameter structure of the models itself is allowed to change during training, e.g., the number of parameter grows when using gradient boosting (GB). This makes defining meaningful prior assumptions for parameter spaces difficult or nearly (practically) infeasible. As we usually do not care about single parameters but the complete resulting function, it seems intuitive to directly express our prior knowledge in hypothesis function space by, e.g., specifying the characteristic length scale, periodicity, or smoothness in general. Fortunately, Bayesian inference can also be formulated in function space. In this case, the prior and posterior distributions are stochastic processes (SPs). The most prominent representative is the Gaussian process (GP), for which the posterior GP can be analytically computed. However, training GPs scale cubically in the number of observations, and the implicit Gaussian likelihood assumption is often violated in reality. In this paper, we introduce Stein functional variational gradient descent (SFVGD). This method provides a general gradient descent method in function space that enables practitioners to train ML models to approximate the posterior SP, assuming certain regularity conditions of the prior SP and the likelihood function hold.

1.1 RELATED WORK

Kernelized Stein Methods These methods combine Stein's identity with a reproducing kernel Hilbert space (RKHS) assumption. Based on a finite particle set, they can either be used to find the optimal transport direction to match a target density or to estimate the score gradient of the empirical distribution of the particles. The former is called Stein variational gradient descent (SVGD) Liu & Wang (2016), and approaches of the latter category are called (non-parametric) score estimators (Zhou et al., 2020). Our method internally uses SVGD and forms a natural analogue in function space. Several extensions to SVGD exist, e.g., approaches incorporating second-order information such as Leviyev et al. (2022) and the more general matrix-kernel valued approach by Wang et al. (2019a). While these extensions usually outperform SVGD, their computational costs are also higher.

Bayesian Neural Networks (BNNs) Typically, BNNs are NNs with weight priors that are trained via variational inference. The prominent representatives are BNNs using *Bayes by Backprop* (Blundell

et al., 2015) and scalable probabilistic backpropagation (Hernandez-Lobato & Adams, 2015). Recently, Immer et al. (2020) proposed transforming BNNs into generalized linear models with inference based on a Gaussian process equivalent to the model. While Markov Chain Monte Carlo (MCMC) methods often are prohibitively expensive to be used for BNNs, some variants, e.g., Chen et al. (2014) account for noisy gradient evaluations and can be used in this setting. However, MCMC-based methods are still usually employed for relatively low-dimensional problems.

Functional BNNs (FBNNs) Sun et al. (2019) proposed to use functional priors to train BNNs. Training BNNs with our descent is closely related to their method, but while they use a score-based approach for the estimation of the derivative of the Kullback-Leibler divergence D_{KL} between the prior SP and the variational SP, we estimate this derivative directly via SVGD. Wang et al. (2019b) also use SVGD for FBNNs but apply SVGD directly to the D_{KL} between the posterior SP and the variational SP. Furthermore, their work does not show that this in fact maximizes a lower bound for the log marginal likelihood. Recently, Ma & Hernández-Lobato (2021) and Rudner et al. (2021) proposed different FBNN approaches that also build upon the results of Sun et al. (2019), but while their methods are specific to training NN function generators, our method can be used to update a set of particle functions in general.

Repulsive Deep Ensembles Repulsive Deep Ensembles are deep ensembles that incorporate repulsive terms in their gradient update, forcing their members' weights apart. A variety of repulsive terms are presented in (D'Angelo & Fortuin, 2021) and (D'Angelo et al., 2021), outperforming the approach by Wang et al. (2019b). However, these approaches mainly focus on weight priors, and empirical findings also only relate to the weight space. In contrast to our work, functional priors can only be applied if a posterior SP with analytical marginal density exists.

GB with Uncertainty The closest neighbor of our approach applied to GB is the ensemble GB scheme proposed by Malinin et al. (2021), which is based on Bayesian ensembles. In contrast to our functional approach, their method is based on approximating the posterior of the model parameters. Another GB-based method is NGBoost proposed by Duan et al. (2019), which directly learns the predictive uncertainty; however, prior knowledge can not be taken into account.

1.2 Our contribution

We propose a novel natural extension of SVGD in function space (Section 3), which enables the practitioner to match a target SP. This approach can be implemented in a BNN or as GB routine (Section 3.3). Using real-world benchmarks, we show that the resulting generator training algorithm is competitive while having less computational costs than the approach of Sun et al. (2019). In contrast to other existing uncertainty-aware GB algorithms, a GB ensemble, when trained via SFVGD, can naturally incorporate prior functional information. These versatile applications of our framework are made possible by providing a unifying view of NNs and GB from a functional analysis perspective.

2 Background

2.1 Supervised ML from a functional analysis perspective

Given a labeled dataset $\mathcal{D} \in (\mathcal{X} \times \mathcal{Y})^n \sim \mathbb{P}_{\mathbf{x}y}^n$ of $n \in \mathbb{N}$ independent and identically distributed (i.i.d.) observations from an unknown data generating process $\mathbb{P}_{\mathbf{x}y}$, a supervised ML algorithm tries to construct a risk optimal model f under a pre-specified loss L. In this case, the function f defines a mapping from the feature space $\mathcal{X}$ to the target space $\mathcal{Y}$. The learning algorithm $\mathcal{I}$ to construct f is a function mapping from the set of all datasets $\bigcup_{n\in\mathbb{N}}(\mathcal{X} \times \mathcal{Y})^n$ to a hypothesis space $\mathcal{H}$, which is a subset of the set of all functions mapping from $\mathcal{X}$ to the model output space[1] $\widetilde{\mathcal{Y}} \subset \mathbb{R}^g$ with $g \in \mathbb{N}$. In order to specify the goodness-of-fit of a function f, one can define a loss function $L : \mathcal{Y} \times \widetilde{\mathcal{Y}} \to \mathbb{R}, (y, f(\mathbf{x})) \mapsto L(y, f(\mathbf{x}))$, which measures how well the output of a fixed model $f \in \mathcal{H}$ fits an observation $(\mathbf{x}, y) \sim \mathbb{P}_{\mathbf{x}y}$. In the following, we present supervised ML from a functional analysis perspective. Here, we fix the observation and associate the loss L with the loss functional$L_{(\mathbf{x},y)}[f] : \mathcal{H} \to \mathbb{R}, f \mapsto L(y, f(\mathbf{x}))$. Based on this loss functional, we can define the risk functional of a model f,

$$\mathcal{R}[f] = \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}_{\mathbf{x}y}} L_{(\mathbf{x},y)}[f], \tag{1}$$

[1] If $\mathcal{Y}$ is numeric, $\widetilde{\mathcal{Y}} = \mathcal{Y}$. Otherwise, $\widetilde{\mathcal{Y}}$ is a numerical encoding of $\mathcal{Y}$.

which measures the expected loss of f, and is used to theoretically identify optimal models. In the following, we will assume that the expectation in Eq. (1) exists and is finite. If we knew the usually unknown data generating process and hence the risk functional, we could update any model $f \in \mathcal{H}$ in the direction of the steepest descent in $\mathcal{H}$ w.r.t. $\mathcal{R}$ by following the negative functional gradient of $\mathcal{R}$. The negative functional gradient of $\mathcal{R}$, $-\nabla_f \mathcal{R}[f]$, is itself a mapping from $\mathcal{X}$ to $\widetilde{\mathcal{Y}}$. For every input location $\mathbf{x}$, this gradient returns the direction in model output space $\widetilde{\mathcal{Y}}$, which points to the locally steepest descent w.r.t. $\mathcal{R}$. In the following, unless otherwise stated, the functional derivative is taken in the L^2 space.

Proposition 2.1 *Assuming sufficient regularity and that $L(y, f(\mathbf{x}))$ is partially continuously differentiable w.r.t. $f(\mathbf{x})$, we observe for numeric inputs and model output that*

$$-\nabla_f \mathcal{R}[f](\mathbf{x}) = -p_{\mathbf{x}}(\mathbf{x}) \cdot \mathbb{E}_{y \sim \mathbb{P}_{y|\mathbf{x}}} \frac{\partial L(y, f(\mathbf{x}))}{\partial f(\mathbf{x})}, \tag{2}$$

where $p_{\mathbf{x}}$ is the marginal density of $\mathbf{x}$.

The proof is given in A.1.1. In practice, we usually do not know $p_{\mathbf{x}}$, and since our dataset $\mathcal{D}$ is finite, we only have access to n realizations of $\frac{\partial L(y, f(\mathbf{x}))}{\partial f(\mathbf{x})}$. If the feature space is at least partially continuous, its size $|\mathcal{X}| = \infty$, and we thus cannot estimate $-\nabla_f \mathcal{R}[f](\mathbf{x})$ without additional assumptions. However, we have access to the functional empirical risk $\mathcal{R}_{\text{emp},\mathcal{D}}[f] := \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}} L_{(\mathbf{x}_i, y_i)}[f]$, for which we assume that it converges in mean to $\mathcal{R}$ as $n \to \infty$. Its negative functional gradient can be expressed via the chain rule such that

$$-\nabla_f \mathcal{R}_{\text{emp},\mathcal{D}}[f](\mathbf{x}) = -\sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}} \frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)} \cdot \nabla_f \left[f(\mathbf{x}_i)\right](\mathbf{x}), \tag{3}$$

where $\nabla_f \left[f(\mathbf{x}_i)\right]$ is the functional gradient of the evaluation functional of f at $\mathbf{x}_i$, which evaluates to the Dirac delta function $\delta_{\mathbf{x}_i}$. However, since we take the functional gradient in $\mathcal{H}$, $\nabla_f \left[f(\mathbf{x}_i)\right]$ becomes the projection of $\delta_{\mathbf{x}_i}$ into $\mathcal{H}$. For example, if $\mathcal{H}$ is an RKHS with associated kernel k, then $\nabla_f \left[f(\mathbf{x}_i)\right](\mathbf{x}) = k(\mathbf{x}_i, \mathbf{x})$, i.e., our choice of $\mathcal{H}$ directly influences the "bumpiness" of $\nabla_f \mathcal{R}_{\text{emp},\mathcal{D}}[f]$. Furthermore, we can interpret $\nabla_f \mathcal{R}_{\text{emp},\mathcal{D}}[f]$ as a (jump-)continuous functional representation of the dataset $\partial\mathcal{D}_{L,f} := \{(\mathbf{x}_i, -\frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)}) \big| \ (\mathbf{x}_i, y_i) \in \mathcal{D}\} \subset (\mathcal{X} \times \widetilde{\mathcal{Y}})^n$, which also implicitly defines a learner. In the following, we show how two core supervised ML algorithms (gradient boosting and neural networks) naturally incorporate this functional gradient while training.

Gradient Boosting (GB) For GB (Friedman, 2001), the situation is usually reversed, and we choose a (base) learner $\mathcal{I}_b$ that implicitly defines $\mathcal{H}$ and with which we fit a model to the data set $\partial\mathcal{D}_{L,f}$. GB uses these approximations of the negative functional gradient of the empirical risk to successively update a model $f^{[0]}$ such that

$$f^{[t+1]} = f^{[t]} + \eta^{[t]} b^{[t]} \text{ with } b^{[t]} = \mathcal{I}_b(\partial\mathcal{D}_{L,f^{[t]}}), \tag{4}$$

where $\eta^{[t]} \in \mathbb{R}_{>0}$ is the learning rate and possibly depends on the iteration $t \in \mathbb{N}$. For further details see Appendix (A.2).

Neural Networks (NNs) If f is an NN with parameters $\boldsymbol{\phi}$, then the parameter gradients w.r.t. the empirical risk functional needed for backpropagation can be obtained via the chain rule such that

$$\nabla_{\boldsymbol{\phi}} \mathcal{R}_{\text{emp},\mathcal{D}}[f] = \int_{\mathcal{X}} \nabla_f \mathcal{R}_{\text{emp},\mathcal{D}}[f](\mathbf{x}) \cdot \nabla_{\boldsymbol{\phi}} f(\mathbf{x}) \mathrm{d}\mathbf{x} = \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}} \frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)} \cdot \nabla_{\boldsymbol{\phi}} f(\mathbf{x}_i), \tag{5}$$

where the second equality holds, since here we do not restrict $\mathcal{H}$, i.e., $\nabla_f \left[f(\mathbf{x}_i)\right] = \delta_{\mathbf{x}_i}$.

However, these procedures only assure that we can find an optimal model $f \in \mathcal{H}$ w.r.t. $\mathcal{R}_{\text{emp}}$, which does not imply that f is optimal w.r.t. $\mathcal{R}$. In practice, we tune the hyperparameters of the algorithms – i.e., use data withheld from learning for subsequent model selection – and apply early stopping to find a model f approximately optimal w.r.t. $\mathcal{R}$.

2.2 STOCHASTIC PROCESSES

In this section, we will shortly introduce stochastic processes (SPs) that can be used to represent distributions over functions and thereby allow us to express the uncertainty of models independent of their specific parameter structure. We will regard $\mathcal{X}$ as an index set and let $(\mathcal{Y}, \mathcal{G})$ be a measurable space with σ-algebra $\mathcal{G}$ on the state space $\mathcal{Y}$. For $\mathbf{x} \in \mathcal{X}$ and a given probability space $(\Omega, \mathcal{F}, \mathbb{P})$ based on the sample space Ω, $\mathcal{F}$ is a σ-algebra on Ω and probability measure $\mathbb{P}$. Let $Q(\mathbf{x})$ be a random variable projecting from Ω to $\mathcal{Y}$. An SP Q is the family $\{Q(\mathbf{x});\ \mathbf{x} \in \mathcal{X}\}$ of all random variables $Q(\mathbf{x})$ (Lamperti, 1977). With this, we can define a sample function $f_\omega : \mathcal{X} \to \mathcal{Y}, \mathbf{x} \mapsto Q(\mathbf{x})(\omega)$ for a fixed $\omega \in \Omega$. Often, it is easier to look at SPs from this sample function view: For every $A \in \mathcal{F}$, a set of functions $\{f_\omega;\ \omega \in A\}$ with an associated measure $\mathbb{P}(A)$ can be identified – i.e., SPs define a distribution over functions projecting from $\mathcal{X}$ to $\mathcal{Y}$. For a finite index set $\mathbf{X} := \mathbf{x}_{1:m} \in \mathcal{X}^m$, we denote the finite-dimensional marginal joint distribution over function values $\{Q(\mathbf{x}_1), \dots, Q(\mathbf{x}_m)\}$ as $Q_\mathbf{X}$. In the following, we assume that for every $Q_\mathbf{X}$ exists a corresponding density function $p_{Q_\mathbf{X}} : \mathcal{Y}^m \to \mathbb{R}_{\geq 0}, \mathbf{f}^\mathbf{X} \mapsto p_{Q_\mathbf{X}}(\mathbf{f}^\mathbf{X})$, where $\mathbf{f}^\mathbf{X} := (f(\mathbf{x}_1), \dots, f(\mathbf{x}_m))$ are the function values at $\mathbf{x}_{1:m}$ based on a sample function f where we suppressed the ω to ease the following notation. We will denote the associated functional to this density function with $p_{Q_\mathbf{X}}[f]$.

The D_{KL} is a measure of distance between two distributions over the same probability space. Since SPs are distributions over functions, the D_{KL} can also be used for distances between two SPs. Unfortunately, computing this quantity is non-trivial (Matthews et al., 2015). However, for two consistent and ergodic SPs Q and P, i.e., Q and P can be characterized by marginals over all finite index sets (e.g., GPs), Sun et al. (2019) showed that the D_{KL} between these SPs can be solely expressed in terms of their marginals, i.e.,

$$D_{\mathrm{KL}}(Q||P) = \sup_{m \in \mathbb{N}, \mathbf{X} \in \mathcal{X}^m} D_{\mathrm{KL}}(Q_\mathbf{X} \| P_\mathbf{X}). \tag{6}$$

This expression enables us to find a differentiable distance measure between two stochastic processes.

2.3 STEIN VARIATIONAL GRADIENT DESCENT

SVGD (Liu & Wang, 2016) is a variational Bayesian method. Variational methods can be used to approximate the generally intractable posterior density of a continuous random variable $\boldsymbol{\theta}$

$$p_{\theta|\mathcal{D}}(\boldsymbol{\theta}) = \frac{p_{\mathcal{D}|\boldsymbol{\theta}}(\mathcal{D}|\boldsymbol{\theta}) p_{\boldsymbol{\theta}}(\boldsymbol{\theta})}{\int p_{\mathcal{D}|\boldsymbol{\theta}}(\mathcal{D}|\boldsymbol{\theta}) p_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}}, \tag{7}$$

where $p_{\mathcal{D}|\boldsymbol{\theta}}$ and $p_{\boldsymbol{\theta}}$ are the likelihood and the prior density function, respectively. SVGD tries to match the posterior $p_{\boldsymbol{\theta}|\mathcal{D}}$ with a density q represented via a fixed number $r \in \mathbb{N}$ of pseudo-samples – so-called particles – and iteratively updates them by minimizing $D_{\mathrm{KL}}(q \| p_{\boldsymbol{\theta}|\mathcal{D}}) = \int q(\boldsymbol{\theta}) \frac{\log(q(\boldsymbol{\theta}))}{p_{\boldsymbol{\theta}|\mathcal{D}}(\boldsymbol{\theta})} d\boldsymbol{\theta}$. In an RKHS with associated kernel k, the optimal update direction is found by considering the negative functional derivative

$$-\nabla_f D_{\mathrm{KL}}(q_{[T]} \| p_{\boldsymbol{\theta}|\mathcal{D}})\big|_{f=0} = \mathbb{E}_{\boldsymbol{\theta} \sim q}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}|\mathcal{D}}(\boldsymbol{\theta}) k(\boldsymbol{\theta}, \cdot) + \nabla_{\boldsymbol{\theta}} k(\boldsymbol{\theta}, \cdot)\right], \tag{8}$$

where $T(\boldsymbol{\theta}) = \boldsymbol{\theta} + f(\boldsymbol{\theta})$, and $q_{[T]}$ is the density of $\boldsymbol{\theta}' = T(\boldsymbol{\theta})$ when $\boldsymbol{\theta} \sim q$. We can estimate this functional gradient based on the particles in an unbiased manner, as we are able to evaluate the score function of $p_{\boldsymbol{\theta}|\mathcal{D}}$ (i.e., $\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}|\mathcal{D}}$), although $\log p_{\boldsymbol{\theta}|\mathcal{D}}$ might be intractable.

3 STEIN FUNCTIONAL VARIATIONAL GRADIENT DESCENT

In this section, we develop a functional version of SVGD which we will call *Stein functional variational gradient descent* (SFVGD). While SVGD can be used to approximate the posterior distribution of a continuous random variable, SFVGD can be applied when we are interested in the posterior SP $P_{f|\mathcal{D}}$ defined by its Radon-Nikodym derivative (Schervish, 1995) w.r.t. the prior SP P_f,

$$\frac{dP_{f|\mathcal{D}}}{dP_f}[f] = \frac{p_{\mathcal{D}|f}[\mathcal{D}|f]}{\int p_{\mathcal{D}|f}[\mathcal{D}|f] dP_f[f]}, \tag{9}$$

where $p_{\mathcal{D}|f}$ is the likelihood functional, which measures how likely it is to observe $\mathcal{D}$, given a sample function f. In the following, we assume that the posterior $P_{f|\mathcal{D}}$ exists and also that it is an ergodic and consistent SP. Analogously to SVGD, we try to approximate $P_{f|\mathcal{D}}$ with a distribution Q represented by pseudo-samples. However, for SFVGD, these particles are now functions.

3.1 OBJECTIVE FUNCTION

Since analytical solutions for the differential Eq. (9) only exist in special cases (e.g., if the prior P_f is a GP and the likelihood is also Gaussian), we use the D_{KL} between two SPs to formulate an optimization objective. More specifically, the goal of our framework is to construct an approximating measure Q^* for which it holds that

$$Q^* \in \operatorname*{arg\,min}_{Q \in \mathcal{Q}} D_{\mathrm{KL}}(Q \| P_{f|\mathcal{D}}), \tag{10}$$

where $\mathcal{Q}$ is the set of representable variational posterior processes. Here, we represent Q via $r \in \mathbb{N}$ sample functions $f_1, \ldots, f_r$ from Q, which act as pseudo-samples and which we also call particle functions. It can be shown (Matthews et al., 2015) that minimizing Eq. (10) is equivalent to maximizing the functional evidence lower bound (ELBO) $\mathcal{L}_{\mathcal{D}}$, i.e.,

$$Q^* \in \operatorname*{arg\,max}_{Q \in \mathcal{Q}} \underbrace{\mathbb{E}_{f \sim Q}\left[\ell[\mathcal{D}|f]\right] - D_{\mathrm{KL}}(Q \| P_f)}_{=:\mathcal{L}_{\mathcal{D}}(Q)}, \tag{11}$$

where $\ell[\mathcal{D}|f] := \log p_{\mathcal{D}|f}[\mathcal{D}|f]$. The advantage of formulation (11) over (10) is that Eq. (11) only depends on known quantities. In the following, we apply Eq. 6, i.e., the results of Sun et al. (2019) regarding the D_{KL} of ergodic and consistent SPs, yielding

$$Q^* \in \operatorname*{arg\,max}_{Q \in \mathcal{Q}} \inf_{m \in \mathbb{N}, \mathbf{X} \in \mathcal{X}^m} \underbrace{\mathbb{E}_{f \sim Q}[\ell[\mathcal{D}|f]] - D_{\mathrm{KL}}(Q_{\mathbf{X}} \| P_{f_{\mathbf{X}}})}_{=:\mathcal{L}_{\mathcal{D},\mathbf{X}}(Q)}. \tag{12}$$

In contrast to Sun et al. (2019), however, we do not unfold the D_{KL} term, since we are able to directly take its functional gradient via SVGD. The resulting maximin game formulation of Eq. (12) proves to be challenging to solve, especially since we need to minimize over discrete sets $\mathbf{X}$ and the infimum also does not ensure a finite m. Hence, we follow Sun et al. (2019) by replacing the inner minimization with a sampling-based approach, i.e.,

$$Q^* \in \operatorname*{arg\,max}_{Q \in \mathcal{Q}} \mathbb{E}_{\mathcal{D}_s} \mathbb{E}_{\mathbf{X}_M \sim C_{\mathcal{X}}} \left[\mathcal{L}_{\mathcal{D}_s, [\mathbf{X}_{\mathcal{D}_s}, \mathbf{X}_M]}(Q)\right], \tag{13}$$

where $\mathcal{D}_s$ is a random subsample of size $|\mathcal{D}_s| = s$ drawn from $\mathcal{D}$. $\mathbf{X}_{\mathcal{D}_s}$ are the associated feature vectors of $\mathcal{D}_s$, and $\mathbf{X}_M = [\mathbf{x}_1, \ldots, \mathbf{x}_M]^\top \in \mathcal{X}^M$ are M stacked random feature vectors drawn from a sampling distribution $C_{\mathcal{X}}$ with support $\mathcal{X}$. If $\mathcal{X}$ is bounded, Sun et al. (2019) proposes a uniform distribution for $C_{\mathcal{X}}$. It has been shown in Sun et al. (2019) for $\mathcal{D}_s = \mathcal{D}$ and $M > 1$ that $\mathcal{L}_{\mathcal{D}, [\mathbf{X}_{\mathcal{D}}, \mathbf{X}_M]}$ is a lower bound for the log marginal likelihood $\log p(\mathcal{D})$, i.e., the maximization in Eq. 13 implies the minimization in Eq. 10. Although, as noted by Burt et al. (2020), if $\mathcal{Q}$ is a parametric family, the objective is ill-defined, we did not encounter any problems in practice. Also, we could straightforwardly use the grid functional D_{KL} proposed by Ma & Hernández-Lobato (2021), which fixes some of these theoretical shortcomings. However, note that SFVGD itself does not assume $\mathcal{Q}$ to be parametric.

3.2 FUNCTIONAL DERIVATIVE OF THE OBJECTIVE

When using conventional gradient descent methods, we want to apply a map to update the parameters of our model such that our loss is reduced. In SFVGD, we proceed in a similar manner but update functions towards a loss-minimizing direction. A map that takes a function as an argument and returns another function is called an operator. Hence, we want to express how our objective value Eq. 13 changes when an operator $F : \mathcal{H} \to \mathcal{H}, f \mapsto f_F$ is applied to every $f \sim Q$. This means that the objective value changes with F such that

$$\mathcal{L}_{\mathcal{D}_s, \mathbf{X}}(Q_{[T]}) = \mathbb{E}_{\widetilde{f} \sim Q_{[T]}} \ell[\mathcal{D}_s|f] - D_{\mathrm{KL}}(Q_{[T]\mathbf{X}} \| P_{f_{\mathbf{X}}}), \tag{14}$$

where $T(f) = f + F(f)$ and $Q_{[T]}$ is the distribution of $\widetilde{f} = T(f)$ when $f \sim Q$. Naturally, we are interested in the functional derivative of Eq. 14 w.r.t. to F, since this gives us the direction of the steepest ascent in operator space regarding the functional ELBO. However, in order to make our computations tractable, we must limit the space of feasible operators:

Definition 3.1 *Let $F : \mathcal{H} \to \mathcal{H}, f \mapsto f_F$ be a continuous operator with the property that for all $m \in \mathbb{N}$ and each $\mathbf{X} \in \mathcal{X}^m$ exists a function $F_{\mathbf{X}} : \mathcal{Y}^m \to \mathcal{Y}^m$ such that $\mathbf{f}_F^{\mathbf{X}} = F_{\mathbf{X}}(\mathbf{f}^{\mathbf{X}})$ for any $f \in \mathcal{H}$.We call such an operator "evaluation-only dependent".*

Thus, F does not depend on derivatives of f (which is not a restriction, since we only assumed f to be continuous); we can also treat F as a construction rule of $F_{\mathbf{X}}$ for arbitrary m and $\mathbf{X}$. Now, we can state the functional gradient of the objective functional w.r.t. an evaluation-only dependent operator F, for $\mathbf{X} = [\mathbf{X}_{\mathcal{D}_s}, \mathbf{X}_M]$ and $\widetilde{\mathbf{X}} = \mathbf{X}_{\mathcal{D}_s}$

$$\begin{aligned}\nabla_F \mathcal{L}_{\mathcal{D}_s,\mathbf{X}}(Q_{[T]}) &= \nabla_F \mathbb{E}_{\widetilde{f}\sim Q_{[T]}} \ell[\mathcal{D}_s|\widetilde{f}] - \nabla_F D_{\mathrm{KL}}(Q_{[T]_{\mathbf{X}}} \| P_{f_{\mathbf{X}}}) \\ &= \nabla_F \mathbb{E}_{\widetilde{\mathbf{y}}\sim Q_{[T]_{\widetilde{\mathbf{X}}}}} \ell(\mathcal{D}_s, \widetilde{\mathbf{y}}) - \nabla_F D_{\mathrm{KL}}(Q_{[T]_{\mathbf{X}}} \| P_{f_{\mathbf{X}}}),\end{aligned} \tag{15}$$

where we assumed that there exists a log-likelihood function $\ell : \bigcup_{s\in\mathbb{N}} (\mathcal{X} \times \mathcal{Y})^s \times \mathcal{Y}^s \to \mathbb{R}$ such that $\ell[\mathcal{D}_s|f] = \ell(\mathcal{D}_s, \mathbf{f}^{\widetilde{\mathbf{X}}})$ for every $\mathcal{D}_s$. If we set $F = 0$, then T becomes the identity operator, i.e. $Q_{[T]} = Q$. Since we want to iteratively update our particle functions, we must only consider small perturbations around $F = 0$.

Proposition 3.1 *For an evaluation-only dependent operator F, the functional derivative of the functional ELBO at $F = 0$ evaluated for a function f*

$$\begin{aligned}\nabla_F \mathcal{L}_{\mathcal{D}_s,\mathbf{X}}(Q_{[T]})\big|_{F=0}(f) = \mathbb{E}_{\widetilde{\mathbf{y}}\sim Q_{\widetilde{\mathbf{X}}}} &\left[\nabla_{\widetilde{\mathbf{y}}} \ell(\mathcal{D}_s, \widetilde{\mathbf{y}}) \cdot \delta_{\widetilde{\mathbf{y}}}(\mathbf{f}^{\widetilde{\mathbf{X}}})\right] \cdot \left[\delta_{\widetilde{\mathbf{X}}_1}(\cdot), \dots, \delta_{\widetilde{\mathbf{X}}_s}(\cdot)\right]^\top \\ + \ \mathbb{E}_{\mathbf{y}\sim Q_{\mathbf{X}}} &\left[\nabla_{\mathbf{y}} \log p_{P_{f_{\mathbf{X}}}}(\mathbf{y}) k_{\mathbf{Y}}(\mathbf{y}, \mathbf{f}^{\mathbf{X}}) + \nabla_{\mathbf{y}} k_{\mathbf{Y}}(\mathbf{y}, \mathbf{f}^{\mathbf{X}})\right] \\ &\cdot \left[\delta_{\mathbf{X}_1}(\cdot), \dots, \delta_{\mathbf{X}_{s+M}}(\cdot)\right]^\top,\end{aligned} \tag{16}$$

where we assume that $\mathcal{H}_{\mathbf{Y}} \subset \{f : \mathcal{Y}^{s+M} \to \mathcal{Y}^{s+M}\}$ is an RKHS with associated kernels $k_{\mathbf{Y}}$.

The proof is given in A.1.2, where we also show the following corollary.

Corollary 3.1.1 *For an evaluation-only dependent operator F, the functional derivative of the functional ELBO at $F = 0$ evaluated for a function f*

$$\begin{aligned}\nabla_F \mathcal{L}_{\mathcal{D}_s,\mathbf{X}}(Q_{[T]})\big|_{F=0}(f) = \mathbb{E}_{\widetilde{\mathbf{y}}\sim Q_{\widetilde{\mathbf{X}}}} &\left[\nabla_{\widetilde{\mathbf{y}}} \ell(\mathcal{D}_s, \widetilde{\mathbf{y}}) \cdot k_{\widetilde{\mathbf{Y}}}(\widetilde{\mathbf{y}}, \mathbf{f}^{\widetilde{\mathbf{X}}})\right] \cdot \left[\delta_{\widetilde{\mathbf{X}}_1}(\cdot), \dots, \delta_{\widetilde{\mathbf{X}}_s}(\cdot)\right]^\top \\ + \ \mathbb{E}_{\mathbf{y}\sim Q_{\mathbf{X}}} &\left[\nabla_{\mathbf{y}} \log p_{P_{f_{\mathbf{X}}}}(\mathbf{y}) k_{\mathbf{Y}}(\mathbf{y}, \mathbf{f}^{\mathbf{X}}) + \nabla_{\mathbf{y}} k_{\mathbf{Y}}(\mathbf{y}, \mathbf{f}^{\mathbf{X}})\right] \\ &\cdot \left[\delta_{\mathbf{X}_1}(\cdot), \dots, \delta_{\mathbf{X}_{s+M}}(\cdot)\right]^\top,\end{aligned} \tag{17}$$

where we assume that $\mathcal{H}_{\mathbf{Y}} \subset \{f : \mathcal{Y}^{s+M} \to \mathcal{Y}^{s+M}\}, \mathcal{H}_{\widetilde{\mathbf{Y}}} \subset \{f : \mathcal{Y}^s \to \mathcal{Y}^s\}$ are RKHSs with associated kernels $k_{\mathbf{Y}}, k_{\widetilde{\mathbf{Y}}}$, respectively.

We call $\nabla_F \mathcal{L}_{\mathcal{D}_s,\mathbf{X}}(Q_{[T]})\big|_{F=0}$ the Stein functional variational gradient operator. It inherits its name from SVGD, which internally is used to find the functional derivative of the D_{KL} term. The key idea of SFVGD is that by updating every particle function $f \sim Q$ via functional gradient descent in the direction of $\nabla_F \mathcal{L}_{\mathcal{D}_s,\mathbf{X}}(Q_{[T]})\big|_{F=0}(f)$, we carry out a gradient step in the distribution space. This increases the current overall functional ELBO value we want to maximize by pulling Q closer to Q^* and consequently also closer to the true posterior stochastic process $P_{f|\mathcal{D}}$.

3.3 ALGORITHMS

Based on the particle functions $f_1, \dots, f_r$, we can find an estimator of Eq. 16

$$\begin{aligned}\widetilde{\nabla}_F \mathcal{L}_{\mathcal{D}_s,\mathbf{X}}(Q_{[T]})\big|_{F=0}(f) = \frac{1}{r}\sum_{i=1}^{r} &\left[\nabla_{\mathbf{f_i}_{\widetilde{\mathbf{X}}}} \ell(\mathcal{D}_s, \mathbf{f_i}^{\widetilde{\mathbf{X}}}) \delta_{\mathbf{f_i}_{\widetilde{\mathbf{X}}}}(\mathbf{f}^{\widetilde{\mathbf{X}}})\right] \cdot \left[\delta_{\widetilde{\mathbf{X}}_1}(\cdot), \dots, \delta_{\widetilde{\mathbf{X}}_s}(\cdot)\right]^\top \\ + \ \frac{\lambda}{r}\sum_{i=1}^{r} &\left[\nabla_{\mathbf{f_i}_{\mathbf{X}}} \log p_{P_{f_{\mathbf{X}}}}(\mathbf{f_i}^{\mathbf{X}}) k_{\mathbf{Y}}(\mathbf{f_i}^{\mathbf{X}}, \mathbf{f}^{\mathbf{X}}) + \nabla_{\mathbf{f_i}_{\mathbf{X}}} k_{\mathbf{Y}}(\mathbf{f_i}^{\mathbf{X}}, \mathbf{f}^{\mathbf{X}})\right] \\ &\cdot \left[\delta_{\mathbf{X}_1}(\cdot), \dots, \delta_{\mathbf{X}_{s+M}}(\cdot)\right]^\top,\end{aligned} \tag{18}$$

Algorithm 1: Stein Functional Variational Gradient Descent Step `sfvgd_step`

Hyperparameters: Dataset $\mathcal{D}$, log likelihood ℓ, prior SP P_f, number of measure points M, sampling distribution $C_{\mathcal{X}}$ over $\mathcal{X}$, regularization parameter λ
Input: Set of particle functions $\{f_i\}_{i=1}^r$ treated as multi-output function f
Output: Input locations to update $\mathbf{X}$, Stein functional variational gradient (of f evaluated at $\mathbf{X}$) $\boldsymbol{\Delta}_{\mathbf{f}\mathbf{X}}$

$\mathbf{X}_M \sim C_{\mathcal{X}}; \mathcal{D}_s = (\widetilde{\mathbf{X}}, \widetilde{\mathbf{y}}) \subset \mathcal{D}$
$\mathbf{X} = \left[\widetilde{\mathbf{X}}, \mathbf{X}_M\right]$
for $j = 1, \ldots, r$ **do**
  $\boldsymbol{\Delta}_{j,\ell} = \frac{1}{r}\sum_{i=1}^r \left[\nabla_{\mathbf{f_i}\widetilde{\mathbf{X}}}\ell(\mathcal{D}_s, \mathbf{f_i}^{\widetilde{\mathbf{X}}})\delta_{\mathbf{f_j}\widetilde{\mathbf{X}}}(\mathbf{f_j}^{\widetilde{\mathbf{X}}})\right] \cdot \left[\delta_{\widetilde{\mathbf{X}}_1}(\cdot), \ldots, \delta_{\widetilde{\mathbf{X}}_s}(\cdot)\right]^\top$
  $\boldsymbol{\Delta}_{j,\mathrm{KL}} = \frac{1}{r}\sum_{i=1}^r \Big[\nabla_{\mathbf{f_i}\mathbf{X}} \log p_{P_{f_\mathbf{X}}}(\mathbf{f_i}^{\mathbf{X}}) k_{\boldsymbol{\Upsilon}}(\mathbf{f_i}^{\mathbf{X}}, \mathbf{f_j}^{\mathbf{X}}) + \nabla_{\mathbf{f_i}\mathbf{X}} k_{\boldsymbol{\Upsilon}}(\mathbf{f_i}^{\mathbf{X}}, \mathbf{f_j}^{\mathbf{X}})\Big]$
    $\cdot \left[\delta_{\mathbf{x}_1}(\cdot), \ldots, \delta_{\mathbf{x}_{s+M}}(\cdot)\right]^\top$
end
$\boldsymbol{\Delta}_{\mathbf{f}\mathbf{X}} = (\boldsymbol{\Delta}_\ell + \lambda \cdot \boldsymbol{\Delta}_{\mathrm{KL}})(\mathbf{X})$

Algorithm 2: Stein Functional Variational Neural Network

Hyperparameters: Same as for `sfvgd_step`
Input: Variational posterior $g(\cdot)$, optimizer `opt`
Output: Variational posterior $g(\cdot)$, which approximates the target distribution
while $\boldsymbol{\phi}$ *not converged* **do**
  $f_i = g(h_{\boldsymbol{\phi}}(\mathbf{X}, \xi_i)), \xi_i \sim p(\xi), \quad i = 1, \ldots, r$
  $\mathbf{X}, \boldsymbol{\Delta}_{\mathbf{f}\mathbf{X}} = \texttt{sfvgd\_step}(f)$
  $\phi = \texttt{opt}(\phi, \mathbf{X}, \boldsymbol{\Delta}_{\mathbf{f}\mathbf{X}})$
end

where we introduce a regularization parameter $\lambda \in \mathbb{R}_{\geq 0}$. Furthermore, if we set $\lambda = 1$, the estimator becomes an unbiased estimator of Eq. (16). Since $\mathcal{L}_{\mathcal{D},\mathbf{X}}$ is a lower bound of the log marginal likelihood $\log p(\mathcal{D})$, it would be preferable to update the particle functions via $\widetilde{\nabla}_F \mathcal{L}_{\mathcal{D},\mathbf{X}}(Q_{[T]})\big|_{F=0}$. However, the major computation bottleneck in Eq. 18 is the calculation of the score gradient $\nabla_{\mathbf{f_i}\mathbf{x}} \log p_{P_{f_\mathbf{X}}}(\mathbf{f_i}^{\mathbf{X}})$ for all particle functions f_i, $i = 1, \ldots, r$ evaluated at $\mathbf{X}$. For example, if P_f is a GP, then the costs of computing $\nabla_{\mathbf{f_i}\mathbf{x}} \log p_{P_{f_\mathbf{X}}}(\mathbf{f_i}^{\mathbf{X}})$ are $\mathcal{O}((s+M)^3 r)$. In addition, the computation of all kernel values $k_{\boldsymbol{\Upsilon}}(\mathbf{f_i}^{\mathbf{X}}, \mathbf{f_j}^{\mathbf{X}})$, $i = 1, \ldots, r$, $j = 1, \ldots, r$ required in Eq. 18 costs $\mathcal{O}(r^2)$. However, this is usually small compared to the cost of computing the score gradient for the functional prior. We choose for M a small constant number, since $\mathcal{L}_{\mathcal{D},[\mathbf{X}_\mathcal{D},\mathbf{X}_M]}$ is a lower bound for the log marginal likelihood $\log p(\mathcal{D})$ for $M > 1$, and we set r to a number of particle functions that can represent the posterior SP reasonably well. Thus, we are interested in estimating $\widetilde{\nabla}_F \mathcal{L}_{\mathcal{D},\mathbf{X}}(Q_{[T]})\big|_{F=0}$ with mini-batches. In principle, an unbiased estimate of $\ell(\mathcal{D}, \mathbf{f_i}^{\mathbf{X}_\mathcal{D}})$ is $n/s \cdot \ell(\mathcal{D}_s, \mathbf{f_i}^{\widetilde{\mathbf{X}}})$, which suggests that $\lambda = s/n$. Although (in general) $\mathcal{L}_{\mathcal{D}_s,\mathbf{X}}$ is not a lower bound of $\log p(\mathcal{D})$, we found in a practice setting that λ to s/n still results in reasonable performance. However, our theoretical framework gives the reassuring guarantee that if we use full-batch training, we would, in fact, maximize a lower bound of $\log p(\mathcal{D})$. In the following, we present two algorithms, namely Stein functional variational NNs and Stein functional variational gradient boosting (A.3.1), based on the estimated Stein functional variational gradient – i.e., they depend on the score gradient of the functional prior evaluated at $\mathbf{X}$. If there exists no analytical score gradient, we can use a score gradient estimator, as suggested in Sun et al. (2019). This only requires function samples of the prior process evaluated at $\mathbf{X}$, but estimating the score gradient is usually computationally expensive (Zhou et al., 2020). Since our approach builds upon SVGD, there exists an additional approach in our framework based on a gradient-free SVGD (Han & Liu, 2018) that only requires the evaluation of the marginal densities of the prior process.

Stein Functional Variational Neural Network (SFVNN) Sun et al. (2019) proposed to train neural networks (NNs) acting as function generators with the negative functional ELBO as loss, which they call Bayesian Functional Variational Neural Networks (BFVNNs). Such a function generator can be modeled via an NN with stochastic weights, which can be represented as a differentiable function $g : \mathcal{Z} \to \mathcal{Y}, z \mapsto g(z)$, where $z \in \mathcal{Z}$ consists of the deterministic input $\mathbf{x}$ and stochastic

inputs, i.e., we can model z as a random variable $z \sim p(z|\mathbf{x})$. These NNs are applicable as long as the reparameterization trick (Kingma & Welling, 2014) can be used, i.e., there exists a random variable $\xi \in \Xi$ with $\xi \sim p(\xi)$ and a differentiable function $h_{\boldsymbol{\phi}} : \mathcal{X} \times \Xi \to \mathcal{Z}$ parametrized by $\boldsymbol{\phi}$ such that $h_{\boldsymbol{\phi}}(\mathbf{x}, \xi) \sim p(z|\mathbf{x})$. With this, we can sample a function by sampling $\xi \sim p(\xi)$ and defining $f_\xi : \mathcal{X} \to \mathcal{Y}, \mathbf{x} \mapsto g(h_{\boldsymbol{\phi}}(\mathbf{x}, \xi))$. In this case, we can write the gradient of Eq. 14 w.r.t. $\boldsymbol{\phi}$ as

$$\begin{aligned}\nabla_{\boldsymbol{\phi}} \mathcal{L}_{\mathcal{D}_s,\mathbf{X}}(Q_{[T]}) = \mathbb{E}_{\xi\sim p(\xi)}\left[\nabla_{\mathbf{f}_\xi^{\widetilde{\mathbf{X}}}} \ell(\mathcal{D}_s, \mathbf{f}_\xi^{\widetilde{\mathbf{X}}}) \nabla_{\boldsymbol{\phi}} \mathbf{f}_\xi^{\widetilde{\mathbf{X}}}\right] \\ - \mathbb{E}_{\xi\sim p(\xi)}\left[\left(\nabla_{\mathbf{f}_\xi^{\mathbf{X}}} \log p_{Q_{\mathbf{X}}}(\mathbf{f}_\xi^{\mathbf{X}}) - \nabla_{\mathbf{f}_\xi^{\mathbf{X}}} \log p_{P_{f_{\mathbf{X}}}}(\mathbf{f}_\xi^{\mathbf{X}})\right) \nabla_{\boldsymbol{\phi}} \mathbf{f}_\xi^{\mathbf{X}}\right].\end{aligned}$$

This is also the result obtained in Sun et al. (2019), where they then use a score estimator (namely, the spectral stein gradient estimator (SSGE; Shi et al., 2018)) to approximate $\nabla_{\mathbf{y}} \log p_{Q_{\mathbf{X}}}(\mathbf{f}_\xi^{\mathbf{X}})$. SSGE estimates the score gradient in an RKHS, i.e., the entropy gradient $\nabla_{\mathbf{f}_\xi^{\mathbf{X}}} \log p_{Q_{\mathbf{X}}}(\mathbf{f}_\xi^{\mathbf{X}})$. Hence, the entropy gradient and the cross entropy gradient $\nabla_{\mathbf{f}_\xi^{\mathbf{X}}} \log p_{P_{f_{\mathbf{X}}}}(\mathbf{f}_\xi^{\mathbf{X}})$ are taken in different functional spaces. Our SFVNN is based on the parameter gradient

$$\nabla_{\boldsymbol{\phi}} \mathcal{L}_{\mathcal{D}_s,\mathbf{X}}(Q_{[T]}) = \mathbb{E}_{\xi\sim p(\xi)}\left[\nabla_F \mathcal{L}_{\mathcal{D}_s,\mathbf{X}}(Q_{[T]})\big|_{F=0}(f_\xi)(\mathbf{X}) \cdot \nabla_{\boldsymbol{\phi}} \mathbf{f}_\xi^{\mathbf{X}}\right] - \underbrace{\mathbb{E}_{\mathbf{f}^{\mathbf{X}}\sim Q_{\mathbf{X}}} \nabla_{\boldsymbol{\phi}} \log p_{Q_{\mathbf{X}},\phi}(\mathbf{f}^{\mathbf{X}})}_{=0},$$

where we use the general Stein functional variational gradient from Eq. 16. In contrast to Sun et al. (2019), we thereby directly take the functional gradient of the D_{KL} term in an RKHS, and our score gradients of the prior process are also subject to the implicit kernel smoothing.

Runtime comparison between SVFNN and FVBNN While FVBNN scales as $\mathcal{O}(r^3 + r^2(s+M))$ (because of SSGE), our approach scales only quadratically in r (because of SVGD), allowing for a larger number of sample functions (see Appendix B).

3.4 Illustrative example

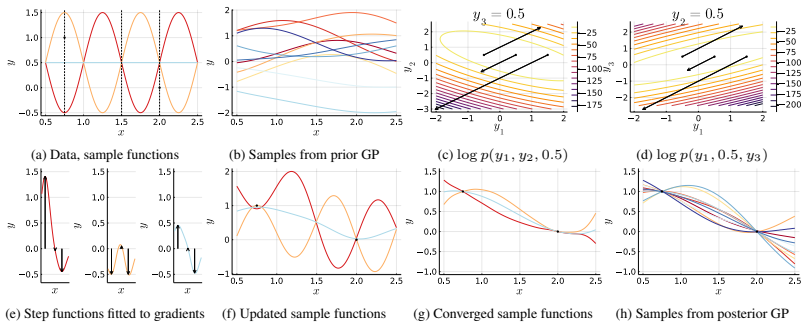

Figure 1: Illustrative example of SFVGB. The points in (a) represent the given data points, and the dashed lines represent all x values that define the marginal prior density in (c,d) w.r.t. the prior GP shown in (b). The arrows in (c) show the resulting SFVGD gradients.

Given three sample functions and two data points $\{(0.75, 1.0), (2.0, 0.0)\}$ (1a), we want to approximate the posterior GP (Figure 1h) w.r.t. the prior GP shown in Figure 1b and a Gaussian likelihood via SFVGB. Hence, we also sample a necessary measure point $x_M = 1.5$. The resulting three-dimensional marginal density is defined by the prior GP. SVGD gives us the optimal update direction for the sample function values to fit this marginal density (Figures 1c, 1d). We fit a kernel ridge regression to these directions after adding the log likelihood gradients at the two data points (Figure 1e). The resulting updated function samples after this SFVGD step can be seen in Figure 1f and the converged function samples in Figure 1g. Qualitatively comparing these converged sample functions in Figure 1g with sample functions from the exact posterior GP in Figure 1h reassures that we are are able to approximate the posterior GP in this toy example reasonably well.

Table 1: Comparison of different methods (columns) on small benchmark data sets (rows) using the average NLL (smaller is better) and RMSE over 10 train-test data splits with standard deviation in brackets. The best performing method for each data set is highlighted in bold.

	Test negative log-likelihood				Test root-mean-square error			
	SFVNN	FVBNN	BNN	GP	SFVNN	FVBNN	BNN	GP
Airfoil	**2.10** (0.17)	2.29 (0.04)	2.62 (0.12)	2.50 (0.14)	**1.82** (0.20)	1.97 (0.19)	3.40 (0.40)	2.77 (0.25)
Concrete	**2.99** (0.19)	3.07 (0.05)	3.25 (0.04)	3.06 (0.05)	**4.58** (0.34)	4.64 (0.54)	6.18 (0.34)	5.13 (0.40)
Diabetes	5.42 (0.08)	5.49 (0.03)	**5.41** (0.04)	6.19 (0.38)	54.57 (3.74)	57.1 (2.48)	**52.7** (2.88)	57.45 (6.6)
Energy	**0.62** (0.10)	0.70 (0.09)	2.26 (0.32)	2.38 (0.05)	0.44 (0.05)	**0.43** (0.08)	2.37 (0.65)	2.34 (0.23)
ForestF	2.38 (0.44)	1.84 (0.05)	**1.83** (0.05)	4.65 (0.45)	1.76 (0.31)	**1.51** (0.07)	**1.51** (0.08)	1.56 (0.08)
Wine	1.96 (1.45)	1.47 (1.07)	-0.03 (0.07)	**-0.04** (0.06)	**0.11** (0.02)	0.14 (0.02)	0.21 (0.03)	0.16 (0.03)
Yacht	**1.06** (0.30)	1.11 (0.24)	1.35 (0.19)	2.86 (0.15)	0.67 (0.26)	**0.61** (0.25)	0.96 (0.28)	3.95 (1.03)
Mean rank	1.86	2.43	2.57	3.14	1.86	1.79	3.07	3.29

Table 2: Comparison of different methods (columns) on large benchmark data sets (rows) using the average NLL (smaller is better) and RMSE over 10 train-test data splits with standard deviation in brackets. The best performing method for each data set is highlighted in bold.

	Test negative log-likelihood			Test root-mean-square error		
	SFVNN	FVBNN	BNN	SFVNN	FVBNN	BNN
GPU	**4.73** (0.04)	4.80 (0.03)	**4.73** (0.02)	27.67 (1.26)	29.6 (0.9)	**27.5** (0.67)
NavalT	**-6.91** (0.06)	-6.85 (0.08)	-5.03 (0.24)	**1.70E-4** (2.5E-5)	1.94E-4 (3.3E-5)	6.50E-4 (6.5E-5)
NavalC	**-6.53** (0.01)	-6.41 (0.05)	-6.44 (0.11)	**1.35E-4** (1.1E-4)	2.39E-4 (3.5E-5)	2.15E-4 (3.9E-5)
Protein	**2.85** (0.01)	2.87 (0.01)	2.96 (0.01)	**4.19** (0.04)	4.27 (0.04)	4.65 (0.05)
VideoMem	11.39 (0.39)	**11.3** (0.10)	11.4 (0.03)	21119 (4910)	**20800** (2320)	21800 (788)
VideoTime	**2.29** (0.05)	2.53 (0.36)	2.86 (1.02)	**2.51** (0.16)	3.14 (0.88)	3.83 (2.08)
Mean rank	1.25	2.17	2.58	1.33	2.17	2.50

4 BENCHMARK STUDY

We further investigate the competitiveness of our approach using its neural network variant (SFVNN) with its closest neighbor, the functional variational Bayesian neural network (FVBNN) from Sun et al. (2019). We also include one well-established BNN baseline (Blundell et al., 2015) and the standard Gaussian Process for the small data sets (where analytical computation is feasible). For results from SVFGB we refer to Section A.3.2 in the Appendix. For most of the datasets, however, the NN provides a better fit to the data. Further details and a contextual bandits experiment can be found in Appendix A.5.

Data and experimental setup All data sets are standardized prior to model fitting and split into 90% training data and 10% test data. For the comparisons, this splitting process is repeated 10 times based on 10 different splits to also evaluate the variability of each method. Further details on data sets, data set-specific pre-processing, and their references can be found in the Appendix.

Details on methods and comparisons In order to provide a fair comparison between methods, we reproduce the best results reported by Sun et al. (2019) for FVBNN and BNN, and also use the same hyperparameters for our method except that while Sun et al. (2019) use $\lambda = 1$, we set λ to s/n. Details for each procedure are given in the Appendix. We compare methods based on the negative log-likelihood (NLL) and the root mean squared error (RMSE) on each test data set and calculate the mean and standard deviation across all 10 data splits.

Results Results are summarized in Tables 1 and 2, indicating that SFVNN is competitive with other existing approaches for both small and large data sets. As the two functional approaches (SFVNN, FVBNN) optimize the same objective, we would expect them to perform similarly, which is confirmed by the results. We further observe that a weight space approach (the BNN) seems to work better than the functional approaches for a few datasets (in particular, for the Wine dataset, where this is expected as the outcome is of discrete nature).

5 CONCLUSION

We introduced a novel gradient descent in distribution space that allows us to update a set of particle functions in a general manner to resemble sample functions from a target process. SFVNN was found to be competitive with or to outperform FBVNN while having less computational costs.

REFERENCES

Rafael Ballester-Ripoll, Enrique G. Paredes, and Renato Pajarola. Sobol tensor trains for global sensitivity analysis, 2017.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1613–1622, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/blundell15.html.

Peter Buehlmann. Boosting for high-dimensional linear models. *The Annals of Statistics*, 34(2): 559–583, 2006.

David R. Burt, Sebastian W. Ober, Adrià Garriga-Alonso, and Mark van der Wilk. Understanding variational inference in function-space, 2020. URL https://arxiv.org/abs/2011.09421.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785. URL http://doi.acm.org/10.1145/2939672.2939785.

Tianqi Chen, Emily B. Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo, 2014. URL https://arxiv.org/abs/1402.4102.

Andrea Coraddu, Luca Oneto, Alessandro Ghio, Stefano Savio, Davide Anguita, and Massimo Figari. Machine learning approaches for improving condition?based maintenance of naval propulsion plants. *Journal of Engineering for the Maritime Environment*, –(–):–, 2014.

Paulo Cortez and Aníbal de Jesus Raimundo Morais. A data mining approach to predict forest fires using meteorological data. 2007.

Francesco D'Angelo and Vincent Fortuin. Repulsive deep ensembles are bayesian, 2021. URL https://arxiv.org/abs/2106.11642.

Francesco D'Angelo, Vincent Fortuin, and Florian Wenzel. On stein variational neural network ensembles, 2021. URL https://arxiv.org/abs/2106.10760.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Tony Duan, Anand Avati, Daisy Yi Ding, Khanh K. Thai, Sanjay Basu, Andrew Y. Ng, and Alejandro Schuler. Ngboost: Natural gradient boosting for probabilistic prediction, 2019. URL https://arxiv.org/abs/1910.03225.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232, 2001. doi: 10.1214/aos/1013203451. URL https://doi.org/10.1214/aos/1013203451.

Jun Han and Qiang Liu. Stein variational gradient descent without gradient. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1900–1908. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/han18b.html.

Jose Miguel Hernandez-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1861–1869, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/hernandez-lobatoc15.html.

Alexander Immer, Maciej Korzepa, and Matthias Bauer. Improving predictions of bayesian neural nets via local linearization, 2020. URL https://arxiv.org/abs/2008.08400.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

John Lamperti. *Stochastic processes : a survey of the mathematical theory / J. Lamperti.* Applied mathematical sciences (Springer-Verlag New York Inc.); v. 23. Springer-Verlag, New York, 1977. ISBN 0387902759.

Alex Leviyev, Joshua Chen, Yifei Wang, Omar Ghattas, and Aaron Zimmerman. A stochastic stein variational newton method, 2022. URL https://arxiv.org/abs/2204.09039.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 2378–2386, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Chao Ma and José Miguel Hernández-Lobato. Functional variational inference based on stochastic process generators. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 21795–21807. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/b613e70fd9f59310cf0a8d33de3f2800-Paper.pdf.

Andrey Malinin, Liudmila Prokhorenkova, and Aleksei Ustimenko. Uncertainty in gradient boosting via ensembles. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=1Jv6b0Zq3qi.

Alexander G. Matthews, James Hensman, Richard E. Turner, and Zoubin Ghahramani. On sparse variational methods and the kullback-leibler divergence between stochastic processes, 2015.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown. In *International Conference on Learning Representations*, 2018.

Tim G. J. Rudner, Zonghao Chen, Yee Whye Teh, and Yarin Gal. Tractable function-space variational inference in bayesian neural networks. 2021.

Mark J. Schervish. *Theory of Statistics*. Springer New York, 1995. doi: 10.1007/978-1-4612-4250-5. URL https://doi.org/10.1007/978-1-4612-4250-5.

Jiaxin Shi, Shengyang Sun, and Jun Zhu. A spectral approach to gradient estimation for implicit distributions, 2018.

Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger Grosse. Functional variational bayesian neural networks, 2019. URL https://arxiv.org/abs/1903.05779.

Athanasios Tsanas and Angeliki Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and buildings*, 49:560–567, 2012.

Dilin Wang, Ziyang Tang, Chandrajit Bajaj, and Qiang Liu. Stein variational gradient descent with matrix-valued kernels, 2019a. URL https://arxiv.org/abs/1910.12794.

Ziyu Wang, Tongzheng Ren, Jun Zhu, and Bo Zhang. Function space particle optimization for bayesian neural networks, 2019b.

I-C Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808, 1998.

Yuhao Zhou, Jiaxin Shi, and Jun Zhu. Nonparametric score estimators. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11513–11522. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/zhou20c.html.

A APPENDIX

A.1 PROOFS

A.1.1 FUNCTIONAL DERIVATIVE OF THE RISK FUNCTIONAL

Assuming that $p(\mathbf{x}, y)L_{(\mathbf{x},y)}[f] \in L^1$ using Fubini's theorem, we find that

$$\mathcal{R}[f] = \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}_{\mathbf{x}y}} L_{(\mathbf{x},y)}[f] = \int_{\mathcal{X}\times\mathcal{Y}} p(\mathbf{x}, y)L_{(\mathbf{x},y)}[f] d\mathbf{x}\, dy \tag{19}$$

$$= \int_{\mathcal{X}} \underbrace{p(\mathbf{x}) \int_{\mathcal{Y}} p(y|\mathbf{x})L_{(\mathbf{x},y)}[f] dy}_{=:\mathcal{L}(\mathbf{x},f(\mathbf{x}))}\, d\mathbf{x}. \tag{20}$$

Assuming that $\mathcal{L}$ is sufficiently smooth using the Euler-Lagrange derivative, we find that

$$\nabla_f \mathcal{R}[f](\mathbf{x}) = \frac{\partial\left(\mathcal{L}(\mathbf{x}, f(\mathbf{x}))\right)}{\partial f(\mathbf{x})} \tag{21}$$

$$= p(\mathbf{x}) \int_{\mathcal{Y}} p(y|\mathbf{x}) \frac{\partial L_{(\mathbf{x},y)}[f]}{\partial f(\mathbf{x})} dy \tag{22}$$

$$= p(\mathbf{x}) \cdot \mathbb{E}_{y\sim\mathbb{P}_{y|\mathbf{x}}} \frac{\partial L_{(\mathbf{x},y)}[f]}{\partial f(\mathbf{x})}. \tag{23}$$

A.1.2 FUNCTIONAL DERIVATIVE OF THE FUNCTIONAL ELBO

Let F be an operator that depends on evaluation only with associated maps $F_{\widetilde{\mathbf{X}}} : \mathcal{Y}^s \to \mathcal{Y}^s, F_{\mathbf{X}} : \mathcal{Y}^{s+M} \to \mathcal{Y}^{s+M}$. Further, let $\mathcal{F}[F] = \mathcal{L}_{\mathcal{D}_s,\mathbf{X}}(Q_{[T]}) = \underbrace{\mathbb{E}_{\widetilde{\mathbf{y}}\sim Q_{[T]\widetilde{\mathbf{X}}}} \ell(\mathcal{D}_s, \widetilde{\mathbf{y}})}_{=\mathcal{F}_1[F_{\widetilde{\mathbf{X}}}]} - \underbrace{D_{\mathrm{KL}}(Q_{[T]\mathbf{X}} \| P_{f_{\mathbf{X}}})}_{=\mathcal{F}_2[F_{\mathbf{X}}]}$.

In general, and under the assumption that ℓ is sufficiently smooth, we find that

$$\nabla_{F_{\widetilde{\mathbf{X}}}} \mathcal{F}_1[F_{\widetilde{\mathbf{X}}}] = \nabla_{F_{\widetilde{\mathbf{X}}}} \mathbb{E}_{\widetilde{\mathbf{y}}\sim Q_{\widetilde{\mathbf{X}}}} \ell(\mathcal{D}_s, \widetilde{\mathbf{y}} + F_{\widetilde{\mathbf{X}}}(\widetilde{\mathbf{y}})) = \mathbb{E}_{\widetilde{\mathbf{y}}\sim Q_{\widetilde{\mathbf{X}}}} \left[\nabla_{\widetilde{\mathbf{y}}} \ell(\mathcal{D}_s, \widetilde{\mathbf{y}} + F_{\widetilde{\mathbf{X}}}(\widetilde{\mathbf{y}})) \cdot \delta_{\widetilde{\mathbf{y}}}(\cdot)\right]. \tag{24}$$

Under the assumption that $\mathcal{H}_{\widetilde{\mathbf{Y}}} \subset \{f : \mathcal{Y}^s \to \mathcal{Y}^s\}$ is an RKHS with associated kernels $k_{\widetilde{\mathbf{Y}}}$, we find that

$$\mathcal{F}_1[F_{\widetilde{\mathbf{X}}} + \varepsilon G_{\widetilde{\mathbf{X}}}] - \mathcal{F}_1[F_{\widetilde{\mathbf{X}}}] = \mathbb{E}_{\widetilde{\mathbf{y}}\sim Q_{\widetilde{\mathbf{X}}}} \left[\ell(\mathcal{D}_s, \widetilde{\mathbf{y}} + F_{\widetilde{\mathbf{X}}}(\widetilde{\mathbf{y}}) + \varepsilon G_{\widetilde{\mathbf{X}}}(\widetilde{\mathbf{y}})) - \ell(\mathcal{D}_s, \widetilde{\mathbf{y}} + F_{\widetilde{\mathbf{X}}}(\widetilde{\mathbf{y}}))\right] \tag{25}$$

$$= \varepsilon\, \mathbb{E}_{\widetilde{\mathbf{y}}\sim Q_{\widetilde{\mathbf{X}}}} \left[\nabla_{\widetilde{\mathbf{y}}} \ell(\mathcal{D}_s, \widetilde{\mathbf{y}} + F_{\widetilde{\mathbf{X}}}(\widetilde{\mathbf{y}})) \cdot G_{\widetilde{\mathbf{X}}}(\widetilde{\mathbf{y}})\right] + \mathcal{O}(\varepsilon^2) \tag{26}$$

$$= \varepsilon\, \mathbb{E}_{\widetilde{\mathbf{y}}\sim Q_{\widetilde{\mathbf{X}}}} \left[\nabla_{\widetilde{\mathbf{y}}} \ell(\mathcal{D}_s, \widetilde{\mathbf{y}} + F_{\widetilde{\mathbf{X}}}(\widetilde{\mathbf{y}})) \cdot \langle k_{\widetilde{\mathbf{Y}}}(\widetilde{\mathbf{y}}, \cdot), G_{\widetilde{\mathbf{X}}}\rangle\right] + \mathcal{O}(\varepsilon^2) \tag{27}$$

$$= \varepsilon\, \langle \mathbb{E}_{\widetilde{\mathbf{y}}\sim Q_{\widetilde{\mathbf{X}}}} \left[\nabla_{\widetilde{\mathbf{y}}} \ell(\mathcal{D}_s, \widetilde{\mathbf{y}} + F_{\widetilde{\mathbf{X}}}(\widetilde{\mathbf{y}})) \cdot k_{\widetilde{\mathbf{Y}}}(\widetilde{\mathbf{y}}, \cdot)\right], G_{\widetilde{\mathbf{X}}}\rangle + \mathcal{O}(\varepsilon^2), \tag{28}$$

from which it follows that

$$\nabla_{F_{\widetilde{\mathbf{X}}}} \mathcal{F}_1[F_{\widetilde{\mathbf{X}}}]\big|_{F_{\widetilde{\mathbf{X}}}=0} = \mathbb{E}_{\widetilde{\mathbf{y}}\sim Q_{\widetilde{\mathbf{X}}}} \left[\nabla_{\widetilde{\mathbf{y}}} \ell(\mathcal{D}_s, \widetilde{\mathbf{y}} + F_{\widetilde{\mathbf{X}}}(\widetilde{\mathbf{y}})) \cdot k_{\widetilde{\mathbf{Y}}}(\widetilde{\mathbf{y}}, \cdot)\right]\big|_{F_{\widetilde{\mathbf{X}}}=0} \tag{29}$$

$$= \mathbb{E}_{\widetilde{\mathbf{y}}\sim Q_{\widetilde{\mathbf{X}}}} \left[\nabla_{\widetilde{\mathbf{y}}} \ell(\mathcal{D}_s, \widetilde{\mathbf{y}}) \cdot k_{\widetilde{\mathbf{Y}}}(\widetilde{\mathbf{y}}, \cdot)\right]. \tag{30}$$

Under the assumption that $\mathcal{H}_{\mathbf{Y}} \subset \{f : \mathcal{Y}^{s+M} \to \mathcal{Y}^{s+M}\}$ is an RKHS with associated kernels $k_{\mathbf{Y}}$, it has been shown in Liu & Wang (2016) that

$$\nabla_{F_{\mathbf{X}}} \mathcal{F}_2[F_{\mathbf{X}}]\big|_{F_{\mathbf{X}}=0} = -\mathbb{E}_{\mathbf{y}\sim Q_{\mathbf{X}}} \left[\nabla_{\mathbf{y}} \log p_{P_{f_{\mathbf{X}}}}(\mathbf{y}) k_{\mathbf{Y}}(\mathbf{y}, \cdot) + \nabla_{\mathbf{y}} k_{\mathbf{Y}}(\mathbf{y}, \cdot)\right]. \tag{31}$$

Using the chain rule, we obtain

$$\nabla_F \mathcal{L}_{\mathcal{D}_s,\mathbf{X}}(Q_{[T]})\big|_{F=0}(f) = \nabla_{F_{\widetilde{\mathbf{X}}}} \mathcal{F}_1[F_{\widetilde{\mathbf{X}}}](\mathbf{f}^{\widetilde{\mathbf{X}}}) \nabla_F \left[F_{\widetilde{\mathbf{X}}}\right](f)\Big|_{F=0} \tag{32}$$

$$- \nabla_{F_{\mathbf{X}}} \mathcal{F}_2[F_{\mathbf{X}}](\mathbf{f}^{\mathbf{X}}) \nabla_F \left[F_{\mathbf{X}}\right](f)\Big|_{F=0}. \tag{33}$$

$$\tag{34}$$

Algorithm 3: Stein Functional Variational Gradient Boosting

Hyperparameters: Same as for `sfvgd_step`
Input: number of iterations $t_{\max}$, learning rate $\eta^{[t]}$ in the t-th iteration, set of initial particle functions $\{f_i^{[0]}\}_{i=1}^r$ treated as multi-output function $f^{[0]}$, multi-output base learner $\mathcal{I}_b$
Output: Set of particle functions $\{f_i^{[t_{\max}]}\}_{i=1}^r$, which approximate the target distribution
for $t = 0, \ldots, t_{\max} - 1$ **do**
  $\mathbf{X}, \boldsymbol{\Delta}_\mathbf{f}\mathbf{x} = \texttt{sfvgd\_step}(f^{[t]})$
  $f^{[t+1]} = f^{[t]} + \eta^{[t]} \cdot \mathcal{I}_b(\mathbf{X}, \boldsymbol{\Delta}_\mathbf{f}\mathbf{x})$
end

Since F is evaluation-only dependent, we observe that $\nabla_F \left[F_{\{\mathbf{x}\}}\right](f) = \delta_\mathbf{x}(\cdot)$ and conclude that

$$\nabla_F \mathcal{L}_{\mathcal{D}_s,\mathbf{X}}(Q_{[T]})\big|_{F=0}(f) = \nabla_{F_{\widetilde{\mathbf{X}}}} \mathcal{F}_1[F_{\widetilde{\mathbf{X}}}](\mathbf{f}^{\widetilde{\mathbf{X}}})\Big|_{F_{\widetilde{\mathbf{X}}}=0} \cdot \left[\delta_{\widetilde{\mathbf{X}}_1}(\cdot), \ldots, \delta_{\widetilde{\mathbf{X}}_s}(\cdot)\right]^\top \tag{35}$$

$$- \nabla_{F_\mathbf{X}} \mathcal{F}_2[F_\mathbf{X}](\mathbf{f}^\mathbf{X})\Big|_{F_\mathbf{X}=0} \cdot \left[\delta_{\mathbf{X}_1}(\cdot), \ldots, \delta_{\mathbf{X}_{s+M}}(\cdot)\right]^\top. \tag{36}$$

If $\mathcal{H}$ is assumed to be an RKHS with associated kernel $k_\mathcal{X}(\mathbf{x}, \cdot)$ then this becomes

$$\nabla_F \mathcal{L}_{\mathcal{D}_s,\mathbf{X}}(Q_{[T]})\big|_{F=0}(f) = \nabla_{F_{\widetilde{\mathbf{X}}}} \mathcal{F}_1[F_{\widetilde{\mathbf{X}}}](\mathbf{f}^{\widetilde{\mathbf{X}}})\Big|_{F_{\widetilde{\mathbf{X}}}=0} \cdot \left[k_\mathcal{X}(\widetilde{\mathbf{X}}_1, \cdot), \ldots, k_\mathcal{X}(\widetilde{\mathbf{X}}_s, \cdot)\right]^\top \tag{37}$$

$$- \nabla_{F_\mathbf{X}} \mathcal{F}_2[F_\mathbf{X}](\mathbf{f}^\mathbf{X})\Big|_{F_\mathbf{X}=0} \cdot \left[k_\mathcal{X}(\mathbf{X}_1, \cdot), \ldots, k_\mathcal{X}(\mathbf{X}_{s+M}, \cdot)\right]^\top. \tag{38}$$

A.2 Gradient Boosting

GB (Friedman, 2001) is a powerful supervised learning algorithm where, iteratively, the residuals are minimized via so-called weak learners. The resulting model consists of a weighted ensemble of these weak learners. In its original form, tree-based learners were used as weak learners, which proved to be effective, especially in the presence of heterogeneous features. XGBoost (Chen & Guestrin, 2016) is a highly efficient algorithm that builds upon the GB paradigm, which proves to be a strong baseline for many structured, supervised regression and classification tasks.

A.3 SFVGB

A.3.1 SFVGB algorithm

We treat the r particle functions mapping from $\mathcal{X}$ to $\mathcal{Y}$ as a single function $f^{[0]}$ mapping from $\mathcal{X}$ to $\mathcal{Y}^r$ – i.e., we identify the i-th sample function $f_i^{[0]}, i = 1, \ldots, r$ with the i-th component of $f^{[0]}$. Analogously to standard GB, we choose a base learner $\mathcal{I}_b$ which defines a hypothesis space $\mathcal{H} \subset \{f : \mathcal{X} \to \mathcal{Y}^r\}$. While this requires the base learner $\mathcal{I}_b$ to be a multi-output learner, it is always possible to use an ensemble of single-output learners. With this, SFVGB is vanilla GB of a multi-output function f where the loss is the negative functional ELBO. Consequently, we update $f^{[t]}$ in the t-th iteration via

$$f^{[t+1]} = f^{[t]} + \eta^{[t]} \mathcal{I}_b\Big(\mathbf{X}, \left[\widetilde{\nabla}_F \mathcal{L}_{\mathcal{D}_s,\mathbf{X}}(Q_{[T]})\big|_{F=0}(f_1^{[t]})(\mathbf{X}), \ldots, \widetilde{\nabla}_F \mathcal{L}_{\mathcal{D}_s,\mathbf{X}}(Q_{[T]})\big|_{F=0}(f_r^{[t]})(\mathbf{X})\right]^\top\Big),$$

using Eq. 18 and a (potentially adaptive) learning rate $\eta^{[t]}$.

A.3.2 SFVGB regression experiments

We also test SFVGB on the small regression datasets and compare it to 1) the approach proposed in Malinin et al. (2021), i.e., uncertainty quantification approaches by randomly subsampling the data in every iteration of a stochastic gradient boosting (SGB) model and 2) boosting generalized linear model (GLMB Buehlmann, 2006) as a baseline approach. As base learners, SGB uses trees and GLMB uses linear models. Our model uses a Nadaraya-Watson kernel regression variant, which does not scale the resulting sum of kernel functions. This is the natural learner if the hypothesis space $\mathcal{H}$ is assumed to be an RHKS, as discussed in section 2.1. The hyperparameters of the SFVGD step are the same as the ones we used for SFVNN. The results after 1000 iterations are summarized in Table 4.

Table 3: Comparison of boosting approaches on the small benchmark data sets (columns) using the average NLL (smaller is better) over 10 train-test data splits with standard deviation in brackets. The best performing method for each data set is highlighted in bold.

	Airfoil	Concrete	Diabetes	Energy	ForestF	Wine	Yacht
SFVGB	2.87 (0.18)	3.41 (0.13)	7.04 (0.67)	2.16 (0.10)	6.31 (0.69)	2.37 (2.00)	3.21 (0.52)
GLMB	3.10 (0.03)	3.93 (0.03)	5.50 (0.03)	3.28 (0.01)	1.84 (0.07)	0.72 (0.02)	3.80 (0.04)
SGB	1.98 (0.05)	3.06 (0.10)	5.51 (0.07)	0.79 (0.49)	1.86 (0.08)	0.11 (0.44)	0.36 (0.23)

Table 4: Comparison of boosting approaches on the small benchmark data sets (columns) using the average RMSE (smaller is better) over 10 train-test data splits with standard deviation in brackets. The best performing method for each data set is highlighted in bold.

	Airfoil	Concrete	Diabetes	Energy	ForestF	Wine	Yacht
SFVGB	3.31 (0.26)	6.71 (0.52)	57.8 (0.67)	1.93 (0.14)	1.53 (0.07)	0.18 (0.04)	4.83 (1.24)
GLMB	4.85 (0.27)	10.5 (1.03)	53.3 (3.42)	3.07 (0.21)	1.51 (0.09)	0.27 (0.04)	8.51 (1.18)
SGB	2.06 (0.21)	5.06 (0.53)	57.6 (2.71)	0.57 (0.09)	1.52 (0.09)	0.29 (0.14)	0.85 (0.45)

Further experimental details Model tuning of SGB is done as explained in Malinin et al. (2021). Here, different tree depths $\in \{3, 4, 5, 6\}$, learning rates $\{0.001, 0.01, 0.1\}$, and numbers of samples $\in \{0.25, 0.5, 0.75\}$ of approaches are trained on the first 80% of the training data and evaluated on the latter 20%. The GLMB approach is tuned using a 10-fold cross-validation to determine the number of stopping iterations, which is the only hyperparameter of the model.

Results Results show that the SFVGB can often improve over the GLMB baseline but still yields inferior performance on some data sets. This is, in particular, the case if there is a rather discrete outcome space (e.g., Wine which only consists of values 0, 1, and 2). The SGB model, in turn, works better than both the SFVGB and GLMB in most cases. This is likely due to the much more flexible base learner structure (as SGB like most of the state-of-the-art boosting approaches uses trees, whereas GLMB uses linear regression and ours uses kernel regression).

A.4 Numerical experiments: further details

A.4.1 Further data details

We use the benchmark data setup proposed by Hernandez-Lobato & Adams (2015) and Sun et al. (2019) for evaluating probabilistic regression approaches. This setup includes selected data sets from the UCI repository – namely, the four smaller data sets Concrete, Energy, Wine, Yacht, and the four larger data sets Naval, Protein, Video (Memory and Time), and GPU. In addition to Sun et al. (2019), we also compare our approaches on three additional smaller data sets (Airfoil, Diabetes, Forest Fire) and further investigate the second task on the Naval data set (i.e., we examine both the compressor decay and the turbine decay state coefficient, referred to as NavalC and NavalT, respectively). In Table 5, the data characteristics and pre-processing steps are listed.

A.4.2 Further experimental details

BNN approaches are fitted using the recommended architecture and tuning parameters by Sun et al. (2019). We reduced the epochs from 10,000 to 1,000 epochs for the smaller data sets to reduce the computational runtime. This did not negatively impact the BNN's performance. In all our benchmark experiments, we follow the setup for BNNs and FVBNNs described by Sun et al. (2019).

A.5 Further results

Here, we provide further results using the application of probabilistic methods for contextual bandits.

Contextual Bandits One important application of uncertainty-aware models is for exploration, as in Bayesian optimization, reinforcement learning, or bandits. Following Sun et al. (2019), we evaluate SFVNN using the contextual bandits benchmark by Riquelme et al. (2018) by re-runing the settings investigated in Sun et al. (2019) and report the cumulative regret based on the best expected

Table 5: Data set characteristics, additional pre-processing and references.

Dataset	# Obs.	# Feat.	Pre-processing	Reference
Airfoil	1503	5	-	Dua & Graff (2017)
Concrete	1030	8	-	Yeh (1998)
Diabetes	442	10	-	Dua & Graff (2017)
Energy	768	8	-	Tsanas & Xifara (2012)
ForestF	517	12	logp1 transformation for `area`; numerical representation for `month` and `day`	Cortez & Morais (2007)
Wine	178	13	-	Dua & Graff (2017)
Yacht	308	6	-	Dua & Graff (2017)
GPU	241600	14	only use run 1 as outcome	Ballester-Ripoll et al. (2017)
NavalT	11934	15	drop features with zero variance	Coraddu et al. (2014)
NavalC	11934	15	drop features with zero variance	Coraddu et al. (2014)
Protein	45730	9	-	Dua & Graff (2017)
Video	68784	19	drop highly correlated features; drop `id` and `b_size`; use dummy-coding for `codec` and `o_codec`	Dua & Graff (2017)

Table 6: Relative contextual bandits regret (relative to the cumulative regret of Uniform sampling) for different data sets (columns) and methods (rows). Numbers in brackets of methods indicate the network sizes. Reported numbers are the mean (and standard derivation in brackets) over 5 trials. The best algorithms per data set are highlighted in bold.

	Adult	Census	Covertype	Jester	Mushroom	Statlog	Wheel
BNN (50)	95.79 (1.33)	66.12 (5.10)	61.18 (1.91)	83.70 (3.31)	7.81 (9.39)	29.2 (4.05)	82.94 (22.2)
BNN (500)	99.16 (1.47)	94.38 (11.9)	74.55 (7.32)	79.08 (5.84)	9.31 (11.1)	64.5 (13.3)	**8.892** (15.0)
BootRMS	82.74 (8.16)	53.40 (18.0)	36.95 (10.3)	63.3 (10.74)	7.11 (5.61)	1.81 (2.70)	119.16 (3.7)
Dropout	85.46 (4.74)	37.27 (12.3)	39.55 (5.84)	65.11 (9.62)	4.43 (6.83)	2.89 (4.26)	28.61 (12.1)
FVBNN (50, 50, 50)	72.84 (11.5)	30.46 (28.4)	**24.29** (14.7)	50.45 (20.0)	2.92 (10.3)	3.46 (3.75)	13.62 (14.7)
FVBNN (50)	75.05 (7.66)	40.65 (14.9)	46.57 (3.99)	57.28 (17.8)	**2.39** (6.24)	**1.58** (2.09)	24.24 (22.0)
ParamNoise	89.00 (5.24)	57.86 (15.5)	48.18 (9.84)	66.52 (13.1)	6.74 (5.89)	7.69 (3.57)	21.93 (15.4)
SFVNN (50, 50, 50)	**71.99** (7.29)	36.65 (30.3)	28.47 (16.4)	**50.09** (21.2)	7.81 (9.35)	4.52 (4.12)	44.12 (27.7)
SFVNN (50)	79.62 (5.44)	**30.05** (13.1)	29.24 (11.4)	55.61 (18.0)	6.88 (7.99)	4.01 (2.19)	91.95 (33.2)
Uniform Sampling	100.0 (0.00)	100.0 (0.00)	100.0 (0.00)	100.0 (0.00)	100.0 (0.00)	100.0 (0.00)	100.0 (0.00)

reward. Following Riquelme et al. (2018), we use the benchmark data sets Adult, Census, Covertype, Financial, Jester, Mushroom, Statlog, and Wheel. For these, the rewards are deterministic, and the regret is equal to the best realized reward. As in previous works, we report the regret as a relative value, relative to a random uniform sampling procedure that emulates Thompson Sampling (see Riquelme et al., 2018). As comparison methods, we use the BNN (with 50 and 500 units), three spinnoffs of the NeuralLinear algorithm (namely, the Bootstrapped NN trained with RMSprop (BootRMS), Parameter Noise (ParamNoise), and Dropout (see Riquelme et al., 2018, for more details)), as well as two variants of the FVBNN (with one and three layers each with 50 units). Experiments are run 5 times with shuffled contexts, for which we report mean and standard deviation of the relative cumulative regret.

Results are given in Table 6, suggesting that both FVBNN and SFVBNN are well- and particularly similar-performing methods in the application of contextual bandits.

B Runtime experiment

B.1 Setup

We trained FVBNN and SVFNN 5 times on the Energy data set for each number of particle functions $r \in \{50, 100, 150\}$ and plotted the resulting runtimes in Figure 2. It becomes apparent that the number of particle functions influences the runtime of FVBNN more substantially than SFVNN, as we expect from our computational complexity analysis.

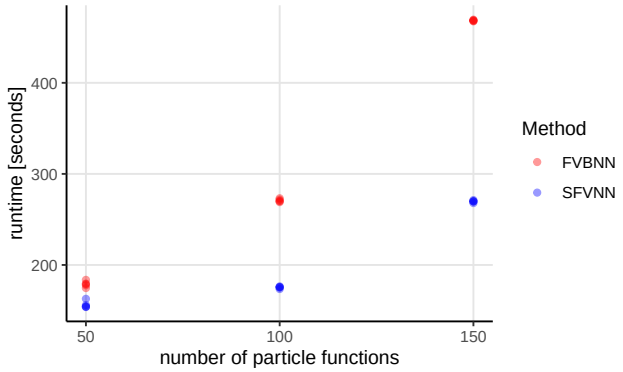

Figure 2: Comparision of the runtimes of FVBNN and SVFNN on the Energy data set with 5 repetitions for each number of particle functions

B.2 Computational environment

All experiments and benchmarks were carried out on an internal cluster with Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz, 32 cores, 64 GB Random-access memory, and operating system Ubuntu 20.04.1 LTS.

Part III.

Contributions to Semi-Implicit Variational Inference

4. Revisiting Unbiased Implicit Variational Inference

Contributing article

Tobias Pielok, Bernd Bischl, and David Rügamer. 2025. Revisiting Unbiased Implicit Variational Inference. In *International Conference on Machine Learning (ICML)*. https://openreview.net/pdf?id=Fm1K8tMlaf.

Copyright information

Copyright © 2025 by the authors and ICML.

Author contributions

Tobias Pielok independently developed the core ideas, theoretical results, and all implementations. David Rügamer worked closely with him on improving the clarity, structure, and overall presentation of the text. Bernd Bischl contributed important feedback on writing, positioning, and benchmarking, helping to sharpen the focus of the paper.

Poster https://icml.cc/media/PosterPDFs/ICML%202025/45871.png?t=1752495528.3460014

Code https://github.com/tpielok/AISIVI

Revisiting Unbiased Implicit Variational Inference

Tobias Pielok [1 2] **Bernd Bischl** [1 2] **David Rügamer** [1 2]

Abstract

Recent years have witnessed growing interest in semi-implicit variational inference (SIVI) methods due to their ability to rapidly generate samples from complex distributions. However, since the likelihood of these samples is non-trivial to estimate in high dimensions, current research focuses on finding effective SIVI training routines. Although unbiased implicit variational inference (UIVI) has largely been dismissed as imprecise and computationally prohibitive because of its inner MCMC loop, we revisit this method and show that UIVI's MCMC loop can be effectively replaced via importance sampling and the optimal proposal distribution can be learned stably by minimizing an expected forward Kullback–Leibler divergence without bias. Our refined approach demonstrates superior performance or parity with state-of-the-art methods on established SIVI benchmarks.

1. Introduction

Bayesian inference, such as sampling-based or variational inference, is an important foundation for constructing uncertainty quantification measures for machine learning models. In variational inference (VI), samples are generated from a target distribution function $p_{\boldsymbol{z}}$ with the associated random variable $\boldsymbol{z}$, which can only be evaluated but not directly sampled from and is possibly unnormalized. This could be, e.g., a Bayesian posterior distribution or the canonical distribution w.r.t. a physical system. For this, a family $\mathcal{Q}_{\boldsymbol{z}}$ over distributions with a tractable sampling procedure is chosen, and a divergence measure D where D quantifies the dissimilarity between two distributions. The target distribution $p_{\boldsymbol{z}}$ can then be approximated by finding $q_{\boldsymbol{z}}^* \in \mathcal{Q}_{\boldsymbol{z}}$ which is closest to $p_{\boldsymbol{z}}$ w.r.t. D, i.e., $q_{\boldsymbol{z}}^* \in \arg\min_{q_{\boldsymbol{z}} \in \mathcal{Q}_{\boldsymbol{z}}} D(q_{\boldsymbol{z}}, p_{\boldsymbol{z}})$.

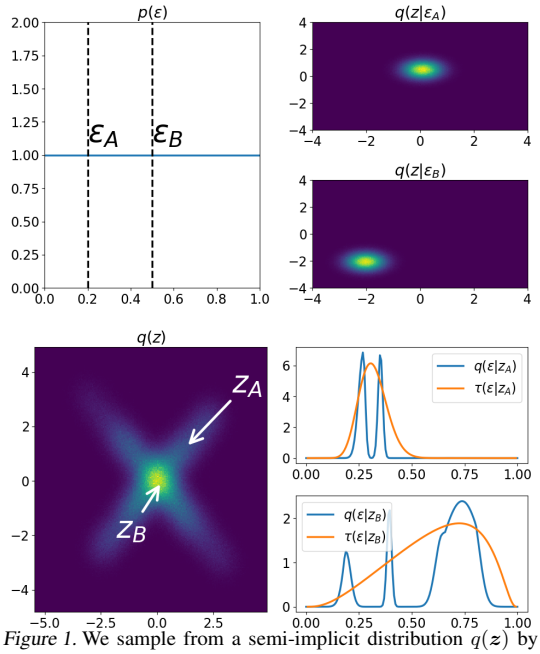

Figure 1. We sample from a semi-implicit distribution $q(\boldsymbol{z})$ by sampling from the latent distribution $p(\boldsymbol{\epsilon})$ and subsequently from the conditional distribution $q(\boldsymbol{z}|\boldsymbol{\epsilon})$. The simple distributions $p(\boldsymbol{\epsilon})$ and $q(\boldsymbol{z}|\boldsymbol{\epsilon})$ can induce a complicated distribution $q(\boldsymbol{z})$ and consequently a potentially even more complicated reverse conditional distribution $q(\boldsymbol{\epsilon}|\boldsymbol{z})$. AISIVI learns a mass-covering representation $\tau(\boldsymbol{\epsilon}|\boldsymbol{z})$ of $q(\boldsymbol{\epsilon}|\boldsymbol{z})$ to estimate $\nabla_{\boldsymbol{z}} \log q(\boldsymbol{z})$ in high dimensions.

1.1. Implicit Variational Inference

In contrast to VI, where we assume that $q_{\boldsymbol{z}} \in \mathcal{Q}_{\boldsymbol{z}}$ is an explicit distribution, i.e., we can evaluate $q_{\boldsymbol{z}}$, for implicit VI (IVI) we have no direct access to $q_{\boldsymbol{z}}$ and can only produce samples from $q_{\boldsymbol{z}}$, i.e., $q_{\boldsymbol{z}}$ is an implicit distribution. Representative examples of explicit and implicit distributions are normalizing flows (NFs) and neural samplers, which transform a random variable via an arbitrary neural network (NN), respectively. While NFs can be trained stably, they are known to smooth out sharp target distributions. In contrast, neural samplers can model highly complex and sharp distributions but are notoriously hard to train. This naturally suggests combining them.

[1]Department of Statistics, LMU Munich, Munich, Germany [2]Munich Center for Machine Learning (MCML), Munich, Germany. Correspondence to: Tobias Pielok <tobias.pielok@stat.uni-muenchen.de>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

Semi-implicit variational inference (SIVI; Yin & Zhou, 2018) offers a compromise between VI and IVI. Since we sample from semi-implicit distribution $q_{\boldsymbol{z}}$ by sampling the parameters $\boldsymbol{y}$ of an explicit distribution[1] $q_{\boldsymbol{z}|\boldsymbol{y}}$ from an implicit distribution $q_{\boldsymbol{y}}$, its representative capabilities come close to those of an implicit distribution, but $q_{\boldsymbol{z}}$ of a semi-implicit distribution can be estimated in a principle manner.

More formally, assuming that the target $\boldsymbol{z} \sim p_{\boldsymbol{z}}$ is a continuous random variable taking values in $Z \subseteq \mathbb{R}^{d_Z}$ where $d_Z \in \mathbb{N}$, we approximate its probability density function via an uncountable mixture of densities s.t.

$$q_{\boldsymbol{z}}(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{y}\sim q_{\boldsymbol{y}}}\left[q_{\boldsymbol{z}|\boldsymbol{y}}(\boldsymbol{z}|\boldsymbol{y})\right]. \tag{1}$$

For SIVI, the random variable $\boldsymbol{y}$ taking values in $Y \subseteq \mathbb{R}^{d_Y}$ where $d_Y \in \mathbb{N}$ is drawn via a neural sampler, i.e.,

$$\boldsymbol{\epsilon} \sim p_{\boldsymbol{\epsilon}} \Rightarrow \boldsymbol{y} = f_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}) \tag{2}$$

where $\boldsymbol{\epsilon}$ is a latent random variable taking values in $E \subseteq \mathbb{R}^{d_E}$ where $d_E \in \mathbb{N}$ and $f_{\boldsymbol{\phi}} : E \to Y$ is a NN with parameters $\boldsymbol{\phi} \in \mathbb{R}^{d_\phi}$ where $d_\phi \in \mathbb{N}$. Consequently, since every $\boldsymbol{\phi}$ defines $q_{\boldsymbol{z}}$, the distribution family $\mathcal{Q}_{\boldsymbol{z}}$ is also parametrized by the NN parameters $\boldsymbol{\phi}$. With Eq. 1 and Eq. 2, we also directly get that

$$q_{\boldsymbol{z}}(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{\epsilon}\sim p_{\boldsymbol{\epsilon}}}\left[q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon})\right] \tag{3}$$

where $q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}) = q_{\boldsymbol{z}|\boldsymbol{y}}(\boldsymbol{z}|f_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}))$.

1.2. Our Contributions

In this work, we focus on the efficient estimation of the score gradient $\nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}}$, which enables us to train SIVI models even in high dimensions. For this, we propose using importance sampling (IS) with an adaptively informed proposal distribution $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ modeled by a conditional normalizing flow (CNF). We show that $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}} = q_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ debiases our score gradient estimate and propose a stable training routine of the CNF via an expected forward Kullback-Leibler divergence. Our contribution advances both mathematical insights of SIVI and contributes two new algorithms.

2. Background and Missed Opportunities

2.1. Reparametrizable Semi-implicit Distributions

In this work, we assume[2] that the reparametrization trick (Kingma & Welling, 2014) is applicable to $q_{\boldsymbol{z}|\boldsymbol{y}}$, i.e., there exist a random variable $\boldsymbol{\eta}$ taking values in $H \subset \mathbb{R}^{d_H}$ where $d_H \in \mathbb{N}$ and a differentiable function $g : Y \times H \to Z$ s.t.

$$\boldsymbol{\epsilon}, \boldsymbol{\eta} \sim p_{\boldsymbol{\epsilon},\boldsymbol{\eta}} \Rightarrow \underbrace{g(f_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}), \boldsymbol{\eta})}_{=:h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon},\boldsymbol{\eta})} \sim q_{\boldsymbol{z}} \tag{4}$$

where $p_{\boldsymbol{\epsilon},\boldsymbol{\eta}}$ is the joint distribution of the independent random variables $\boldsymbol{\epsilon}$ and $\boldsymbol{\eta}$ which does not depend on $\boldsymbol{\phi}$. From this, it directly follows that

$$\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{z}}}\left[a_{\boldsymbol{\phi}}(\boldsymbol{z})\right] = \mathbb{E}_{\boldsymbol{\epsilon},\boldsymbol{\eta}\sim p_{\boldsymbol{\epsilon},\boldsymbol{\eta}}}\left[a_{\boldsymbol{\phi}}(h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon},\boldsymbol{\eta}))\right] \tag{5}$$

where $a_{\boldsymbol{\phi}}(\boldsymbol{z}) : Z \to \mathbb{R}$ is a differentiable function with parameters $\boldsymbol{\phi}$. Hence, under our assumptions, the reparametrization trick can be applied to $q_{\boldsymbol{z}}$.

2.2. Path gradient estimator and D_{KL} minimization

We choose to minimize the reverse Kullback-Leibler divergence D_{KL}, i.e.,

$$D_{\mathrm{KL}}(q_{\boldsymbol{z}}\|p_{\boldsymbol{z}}) = \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{z}}}\left[\log\left(\frac{q_{\boldsymbol{z}}(\boldsymbol{z})}{p_{\boldsymbol{z}}(\boldsymbol{z})}\right)\right]. \tag{6}$$

On the one hand, one of the main advantages of D_{KL} is that if we can evaluate $q_{\boldsymbol{z}}$ we can compute unbiased estimates of the gradients w.r.t. the parameters $\boldsymbol{\phi}$ of $q_{\boldsymbol{z}}$, which is especially useful when stochastic gradient descent methods are employed to minimize the objective. On the other hand, the reverse D_{KL} is known to underestimate the variance if the variational distribution $q_{\boldsymbol{z}}$ is not sufficiently expressive (Andrade, 2024). However, for SIVI, this is rarely relevant, as the variational distribution $q_{\boldsymbol{z}}$ is highly expressive due to its implicit nature.

Since $q_{\boldsymbol{z}}$ is amenable to the reparametrization trick, we can follow Roeder et al. (2017) to formulate a low-variance gradient estimator of D_{KL}, the so-called path gradient estimator

$$\begin{aligned}\nabla_{\boldsymbol{\phi}} D_{\mathrm{KL}}(q_{\boldsymbol{z}}\|p_{\boldsymbol{z}}) = \\ \mathbb{E}_{\boldsymbol{\epsilon},\boldsymbol{\eta}\sim p_{\boldsymbol{\epsilon},\boldsymbol{\eta}}}\Big[\nabla_{\boldsymbol{z}}\left(\log q_{\boldsymbol{z}}(\boldsymbol{z}) - \log p_{\boldsymbol{z}}(\boldsymbol{z})\right)\Big|_{\boldsymbol{z}=h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon},\boldsymbol{\eta})} \\ \cdot \nabla_{\boldsymbol{\phi}} h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon},\boldsymbol{\eta})\Big].\end{aligned} \tag{7}$$

While this result also appeared in the context of SIVI as an intermediate result in Titsias & Ruiz (2019), its far-reaching implications were not discussed since this expression was not of interest for the authors' final derivation (see Section 2.3). Not only does the path gradient estimator in Eq. 7 reduce the variance of the gradient estimation, but it also vastly reduces the computational demand in contrast to the reparametrization trick, for which we would need to estimate the gradient

$$\begin{aligned}\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{z}}(h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon},\boldsymbol{\eta})) = \\ \nabla_{\boldsymbol{\phi}} \log\left[\mathbb{E}_{\widetilde{\boldsymbol{\epsilon}}\sim p_{\boldsymbol{\epsilon}}}\left[q_{\boldsymbol{z}|\boldsymbol{y}}(h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon},\boldsymbol{\eta})|f_{\boldsymbol{\phi}}(\widetilde{\boldsymbol{\epsilon}})\right]\right].\end{aligned} \tag{8}$$

[1] usually a common, unimodal distribution such as, e.g., a normal distribution

[2] We could even lessen our assumption by only assuming that implicit reparametrization gradients can be computed (Figurnov et al., 2018), but this is not the focus of this paper.

This simple observation leads to a surprisingly well-performing approach, which we will introduce in Section 3.

2.3. Unbiased Implicit Variational Inference

The problematic term of the path gradient estimator in Eq. 7 is the score gradient $\nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}}(\boldsymbol{z})$, for which no analytical expression exists. Titsias & Ruiz (2019) proved for UIVI that

$$\mathbb{E}_{\boldsymbol{\epsilon} \sim q_{\boldsymbol{\epsilon}|\boldsymbol{z}}} \left[\nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}) \right] = \nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}}(\boldsymbol{z}), \tag{9}$$

i.e., if we can produce samples from the intractable conditional distribution $q_{\boldsymbol{\epsilon}|\boldsymbol{z}}$, we can compute an unbiased estimate of the score gradient $\nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}}(\boldsymbol{z})$. Titsias & Ruiz (2019) propose to sample $\boldsymbol{z}, \boldsymbol{\epsilon} \sim q_{\boldsymbol{z},\boldsymbol{\epsilon}}$ and use MCMC with target distribution[3] $q_{\boldsymbol{\epsilon}|\boldsymbol{z}}$. The MCMC chains are initialized at $\boldsymbol{\epsilon}$ because it already stems from the stationary distribution $q_{\boldsymbol{\epsilon}|\boldsymbol{z}}$. However, we can not use $\boldsymbol{\epsilon}$ directly since this would violate the independence assumption, which is needed for an unbiased estimate in Eq. 9. Therefore, MCMC has to run as long as the sample produced by the i-th chain $\boldsymbol{\epsilon}_i'$ is independent of $\boldsymbol{\epsilon}$. Titsias & Ruiz (2019) argue that only a few steps of MCMC are needed since the chains are already initialized at the stationary distribution. However, as it can be seen in Figure 1, $q_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ is likely multimodal with regions of vanishing probability potentially occurring between the modes due to the implicit and possibly very complicated nature of $q_{\boldsymbol{z}}$. In such cases, very long chains would be needed to effectively break the dependence between $\boldsymbol{\epsilon}$ and $\boldsymbol{\epsilon}_i'$, rendering the already computationally intensive method as prohibitive. Furthermore, note that the number of chains cannot reduce the bias introduced by the prevailing dependence between $\boldsymbol{\epsilon}$ and $\boldsymbol{\epsilon}_i'$.

In light of these observations, we propose a novel method in Section 3 to fix the encountered shortcomings.

2.4. Conditional Normalizing Flows

Normalizing flows (NF; see, e.g., Papamakarios et al., 2021) leverage the change of variable method to model complex distributions by repeatedly transforming a random variable stemming from a simple error distribution. More specifically, for a random variable $\boldsymbol{u}$ taking values in $U \subseteq \mathbb{R}^{d_U}$ where $d_U \in \mathbb{N}$ and a differentiable and invertible transformation $T_{\boldsymbol{\theta}} : U \to U$ with parameters $\boldsymbol{\theta} \in \Theta \in \mathcal{R}$ it holds that

$$\begin{aligned} \boldsymbol{u} \sim p_{\boldsymbol{u}}, \boldsymbol{\epsilon} = T_{\boldsymbol{\theta}}(\boldsymbol{u}) \Rightarrow \quad & \boldsymbol{\epsilon} \sim q_{\boldsymbol{\epsilon}}, \\ & q_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}) = p_{\boldsymbol{u}}(T_{\boldsymbol{\theta}}^{-1}(\boldsymbol{\epsilon})) \left| \det J_{T_{\boldsymbol{\theta}}^{-1}}(\boldsymbol{\epsilon}) \right| \end{aligned} \tag{10}$$

[3] We know $q_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ up to a normalizing constant, i.e., $q_{\boldsymbol{z},\boldsymbol{\epsilon}}$, which suffices for MCMC

where $J_{T_{\boldsymbol{\theta}}^{-1}}$ is the Jacobian of the inverse function of $T_{\boldsymbol{\theta}}$. A *conditional* NF (CNF) is a differentiable map $T_{\boldsymbol{\theta}} : U \times Z \to U, (\boldsymbol{\epsilon}, \boldsymbol{z}) \mapsto T_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}, \boldsymbol{z})$ such that for every $\boldsymbol{z} \in Z$ it holds that $T_{\boldsymbol{\theta}}(\cdot, \boldsymbol{z})$ is a NF.

A plethora of different NFs have been proposed over the last years. In this work, we use affine coupling layers as introduced in RealNVP (Dinh et al., 2017) because sampling and evaluating their likelihood is equally computationally efficient, and they can be scaled up to high dimensions (Andrade, 2024). Specifically, we use a conditional variant of affine coupling layers similar to one introduced in Lu & Huang (2020). While this is a natural choice, other combinations could provide additional performance gains as also discussed in Section 6.

3. Method

Starting from Eq. 7 we can rewrite the problematic score term, i.e.,

$$\nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}}(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \log \left[\mathbb{E}_{\boldsymbol{\epsilon} \sim p_{\boldsymbol{\epsilon}}} \left[q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}) \right] \right]. \tag{11}$$

Thus, a straightforward Monte Carlo (MC) estimator of the score gradient is

$$s_{\mathrm{MC},k}(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \log \left(\frac{1}{k} \sum_{i=1}^{k} q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i) \right), \tag{12}$$

where $(\boldsymbol{z}, \boldsymbol{\epsilon}_1) \sim q_{\boldsymbol{z},\boldsymbol{\epsilon}}$ and $\boldsymbol{\epsilon}_i \overset{\text{i.i.d.}}{\sim} p_{\boldsymbol{\epsilon}}, i = 2, \ldots, k$. This is a consistent estimator of the score gradient $\nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}}(\boldsymbol{z})$ and for large k its bias

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{\epsilon}_i \sim p_{\boldsymbol{\epsilon}}} \left[\nabla_{\boldsymbol{z}} s_{\mathrm{MC},k}(\boldsymbol{z}) \right] - \nabla_{\boldsymbol{z}} \log \left(q_{\boldsymbol{z}}(\boldsymbol{z}) \right) \approx \\ &\qquad - \nabla_{\boldsymbol{z}} \left(\frac{\mathbb{V}_{\boldsymbol{\epsilon} \sim p_{\boldsymbol{\epsilon}}} \left[q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}) \right]}{2(k-1) \cdot q_{\boldsymbol{z}}(\boldsymbol{z})^2} \right) \end{aligned} \tag{13}$$

(see Appendix A.1 for the proof). Note that including $\boldsymbol{\epsilon}_1$ introduces additional bias but strongly reduces the variance since $\boldsymbol{\epsilon}_1 \sim q_{\boldsymbol{\epsilon}|\boldsymbol{z}}$. A closely related estimator was derived in Molchanov et al. (2019), but their estimator is purely based on the reparametrization trick and does not benefit from the advantages discussed in Section 2.2 and Section 3.2.

Although we would expect that for high dimensions, the contribution of $q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)$ resulting from uninformed $\boldsymbol{\epsilon}_i$ to be nearly negligible to our estimator, $s_{\mathrm{MC},k}$ performs surprisingly well.

Based on the previous observation, we devise a new importance sampling (IS) version of Eq. 11, given as follows:

$$\begin{aligned} &\nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}}(\boldsymbol{z}) = \\ &\qquad \nabla_{\boldsymbol{z}} \log \left(\mathop{\mathbb{E}}_{\boldsymbol{\epsilon} \sim \tau_{\boldsymbol{\epsilon}|\widetilde{\boldsymbol{z}}}} \left[\frac{p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}) q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon})}{\tau_{\boldsymbol{\epsilon}|\widetilde{\boldsymbol{z}}}(\boldsymbol{\epsilon}|\widetilde{\boldsymbol{z}})} \right] \right) \Bigg|_{\widetilde{\boldsymbol{z}}=\boldsymbol{z}}. \end{aligned} \tag{14}$$

The idea of enhancing SIVI with importance sampling was also proposed by Sobolev & Vetrov (2019), but their approach is more expensive than ours due to the joint optimization of the proposal distribution and the SIVI model, rendering more expressive conditional models, such as CNFs, infeasible in practice.

Importance Sampling Estimator Based on Eq. 14, we can estimate $\nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}}(\boldsymbol{z})$ using the following score gradient estimator

$$s_{\mathrm{IS},k}(z) = \nabla_{\boldsymbol{z}} \log \left(\frac{1}{k} \sum_{i=1}^{k} \frac{p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}_i) q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)}{\tau_{\boldsymbol{\epsilon}|\widetilde{\boldsymbol{z}}}(\boldsymbol{\epsilon}_i|\widetilde{\boldsymbol{z}})} \right) \Bigg|_{\widetilde{\boldsymbol{z}}=\boldsymbol{z}}, \quad (15)$$

where $\boldsymbol{\epsilon}_i \sim \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}, i = 1, \dots, k$. We show in Appendix A.2 that this estimator is consistent when $\mathrm{supp}(q_{\boldsymbol{\epsilon}|\boldsymbol{z}}) \subset \mathrm{supp}(\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}})$. To also make this estimator efficient, we need to generate samples $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ and evaluate their likelihood efficiently. A suitable option in this case is to model $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ with a sequence of conditional affine coupling layers (see Section 2.4).

Since we optimize $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ and $q_{\boldsymbol{z}}$ alternately, we are interested in the optimal $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ for a fixed $q_{\boldsymbol{z}}$. This leads us to the following proposition:

Proposition 3.1. *Choosing $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}} = q_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ debiases our proposed score gradient estimate $s_{\mathrm{IS},k}$, i.e.,*

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{\epsilon}_i \sim q_{\boldsymbol{\epsilon}|\boldsymbol{z}}} \nabla_{\boldsymbol{z}} \log \left(\frac{1}{k} \sum_{i=1}^{k} \frac{p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}_i) q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)}{q_{\boldsymbol{\epsilon}|\widetilde{\boldsymbol{z}}}(\boldsymbol{\epsilon}_i|\widetilde{\boldsymbol{z}})} \right) \Bigg|_{\widetilde{\boldsymbol{z}}=\boldsymbol{z}} \quad (16) \\ &\quad = \nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}}(\boldsymbol{z}). \end{aligned}$$

We prove Proposition 3.1 in Section 3.1. Hence, we propose to learn $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ by minimizing the expected forward Kullback-Leibler divergence $\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{z}}} \left[D_{\mathrm{KL}}(q_{\boldsymbol{\epsilon}|\boldsymbol{z}} \| \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}) \right]$, for which we can estimate its gradient w.r.t. to the parameters $\boldsymbol{\theta}$ of the NF without bias since

$$\begin{aligned} &\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{z}}} \left[D_{\mathrm{KL}}(q_{\boldsymbol{\epsilon}|\boldsymbol{z}} \| \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}) \right] && (17) \\ &\quad = \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{z}}} \mathbb{E}_{\boldsymbol{\epsilon}\sim q_{\boldsymbol{\epsilon}|\boldsymbol{z}}} \nabla_{\boldsymbol{\theta}} \log \left(\frac{q_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z})}{\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z})} \right) && (18) \\ &\quad = -\mathbb{E}_{\boldsymbol{z},\boldsymbol{\epsilon}\sim q_{\boldsymbol{z},\boldsymbol{\epsilon}}} \nabla_{\boldsymbol{\theta}} \log \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z}) && (19) \end{aligned}$$

which holds because $q_{\boldsymbol{z},\boldsymbol{\epsilon}}$ does not depend on $\boldsymbol{\theta}$. The following proposition assures the validity of our procedure:

Proposition 3.2. *Minimizing $\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{z}}} \left[D_{\mathrm{KL}}(q_{\boldsymbol{\epsilon}|\boldsymbol{z}} \| \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}) \right]$ is equivalent to minimizing $D_{\mathrm{KL}}(q_{\boldsymbol{z},\boldsymbol{\epsilon}} \| \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}} \cdot q_{\boldsymbol{z}})$.*

This follows from the fact that

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{z}}} \left[D_{\mathrm{KL}}(q_{\boldsymbol{\epsilon}|\boldsymbol{z}} \| \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}) \right] && (20) \\ &\quad = \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{z}}} \mathbb{E}_{\boldsymbol{\epsilon}\sim q_{\boldsymbol{\epsilon}|\boldsymbol{z}}} \log \left(\frac{q_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z})}{\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z})} \right) && (21) \\ &\quad = \mathbb{E}_{\boldsymbol{z},\boldsymbol{\epsilon}\sim q_{\boldsymbol{z},\boldsymbol{\epsilon}}} \log \left(\frac{q_{\boldsymbol{z},\boldsymbol{\epsilon}}(\boldsymbol{z},\boldsymbol{\epsilon})}{\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z}) q_{\boldsymbol{z}}(\boldsymbol{z})} \right) && (22) \\ &\quad = D_{\mathrm{KL}}(q_{\boldsymbol{z},\boldsymbol{\epsilon}} \| \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}} \cdot q_{\boldsymbol{z}}) && (23) \end{aligned}$$

From this, assuming that $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ is sufficiently flexible, it directly follows that at the global optimum $\tau^*_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ of the expected forward D_{KL} it holds that

$$q_{\boldsymbol{z},\boldsymbol{\epsilon}} = \tau^*_{\boldsymbol{\epsilon}|\boldsymbol{z}} \cdot q_{\boldsymbol{z}} \Rightarrow \tau^*_{\boldsymbol{\epsilon}|\boldsymbol{z}} = \frac{q_{\boldsymbol{z},\boldsymbol{\epsilon}}}{q_{\boldsymbol{z}}} = q_{\boldsymbol{\epsilon}|\boldsymbol{z}}. \quad (24)$$

Being of particular importance for understanding our finding, we also include the proof of Proposition 3.1 in the following.

3.1. Proof of Proposition 3.1

First note that

$$\frac{p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}_i)}{q_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}_i|\boldsymbol{z})} = \frac{p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}_i) q_{\boldsymbol{z}}(\boldsymbol{z})}{p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}_i) q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(z|\boldsymbol{\epsilon}_i)} = \frac{q_{\boldsymbol{z}}(\boldsymbol{z})}{q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)}. \quad (25)$$

With this, we get that

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{\epsilon}_i \sim q_{\boldsymbol{\epsilon}|\boldsymbol{z}}} \nabla_{\boldsymbol{z}} \log \left(\frac{1}{k} \sum_{i=1}^{k} \frac{p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}_i) q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)}{q_{\boldsymbol{\epsilon}|\widetilde{\boldsymbol{z}}}(\boldsymbol{\epsilon}_i|\widetilde{\boldsymbol{z}})} \right) \Bigg|_{\widetilde{\boldsymbol{z}}=\boldsymbol{z}} && (26) \\ &= \mathbb{E}_{\boldsymbol{\epsilon}_i \sim q_{\boldsymbol{\epsilon}|\boldsymbol{z}}} \left[\frac{\frac{1}{k} \sum_{i=1}^{k} \frac{p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}_i)}{q_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}_i|\boldsymbol{z})} \nabla_{\boldsymbol{z}} q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)}{\frac{1}{k} \sum_{i=1}^{k} \frac{p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}_i)}{q_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}_i|\boldsymbol{z})} q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)} \right] && (27) \\ &= \frac{\frac{1}{k} \sum_{i=1}^{k} \mathbb{E}_{\boldsymbol{\epsilon}_i \sim q_{\boldsymbol{\epsilon}|\boldsymbol{z}}} \left[\frac{p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}_i)}{q_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}_i|\boldsymbol{z})} \nabla_{\boldsymbol{z}} q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i) \right]}{1/k \sum_{i=1}^{k} q_{\boldsymbol{z}}(\boldsymbol{z})} && (28) \\ &= \frac{\sum_{i=1}^{k} \mathbb{E}_{\boldsymbol{\epsilon}_i \sim q_{\boldsymbol{\epsilon}|\boldsymbol{z}}} \left[\frac{q_{\boldsymbol{z}}(\boldsymbol{z}) q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)}{q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)} \nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i) \right]}{k \cdot q_{\boldsymbol{z}}(\boldsymbol{z})} && (29) \\ &= \frac{1}{k} \sum_{i=1}^{k} \mathbb{E}_{\boldsymbol{\epsilon}_i \sim q_{\boldsymbol{\epsilon}|\boldsymbol{z}}} \left[\nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i) \right] && (30) \\ &\overset{\text{Eq. 9}}{=} \nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}}(\boldsymbol{z}). && (31) \end{aligned}$$

3.2. Training Under Memory Constraints

One of the main advantages of our proposed score gradient estimators $s_{\mathrm{MC},k}$ and $s_{\mathrm{IS},k}$ is that increasing k, i.e., the number of samples $\boldsymbol{\epsilon}_i$, does not increase the computational cost of backpropagation w.r.t. the parameters of our SIVI model $\boldsymbol{\phi}$ because we follow the path gradient. This insight motivates the following procedure, which allows us to train

our SIVI models with constant memory requirements independent of k.

First, note that both our score gradient estimators can be written s.t.

$$s(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \ell(\boldsymbol{z}, \widetilde{\boldsymbol{z}})\Big|_{\widetilde{\boldsymbol{z}}=\boldsymbol{z}} \quad \text{with} \tag{32}$$

$$\ell(\boldsymbol{z}, \widetilde{\boldsymbol{z}}) = \log\left(\frac{1}{k}\sum_{i=1}^{k} w(\boldsymbol{\epsilon}_i|\widetilde{\boldsymbol{z}}) q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)\right), \tag{33}$$

where choosing $w(\boldsymbol{\epsilon}_i|\widetilde{\boldsymbol{z}}) = 1$ or $w(\boldsymbol{\epsilon}_i|\widetilde{\boldsymbol{z}}) = \frac{p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}_i)}{q_{\boldsymbol{\epsilon}|\widetilde{\boldsymbol{z}}}(\boldsymbol{\epsilon}_i|\widetilde{\boldsymbol{z}})}$ results in $s_{\mathrm{MC},k}$ and $s_{\mathrm{IS},k}$, respectively. Since evaluating $q_{\boldsymbol{\epsilon}|\widetilde{\boldsymbol{z}}}(\boldsymbol{\epsilon}_i|\widetilde{\boldsymbol{z}})$ is computationally non-intensive because of the inner neural sampler, we could, in principle, process very large $\boldsymbol{\epsilon}$ batches. However, since our memory is constrained, we need a way to aggregate score gradient estimators computed on different $\boldsymbol{\epsilon}$ batches.

Efficient Aggregation on Batch Level Assume we have computed the score gradient estimates s_1, s_2 with associated log probability density estimates ℓ_1, ℓ_2 of the $\boldsymbol{\epsilon}_i$ batches of sizes $j \cdot b$ and b, respectively, with $j, b \in \mathbb{N}$. Then, we show in Appendix A.3 that if we aggregate these estimates s.t.

$$\begin{aligned}\ell_3(\boldsymbol{z}, \widetilde{\boldsymbol{z}}) = \mathrm{logaddexp}\left(\ell_1(\boldsymbol{z}, \widetilde{\boldsymbol{z}}) + \log j, \ell_2(\boldsymbol{z}, \widetilde{\boldsymbol{z}})\right) \\ - \log(j+1),\end{aligned} \tag{34}$$

and

$$\begin{aligned} s_3(\boldsymbol{z}) &= \alpha_1 s_1(\boldsymbol{z}) + \alpha_2 s_2(\boldsymbol{z}) \text{ with} \\ \alpha_1 &= \exp\left(\ell_1(\boldsymbol{z}, \widetilde{\boldsymbol{z}}) - \ell_3(\boldsymbol{z}, \widetilde{\boldsymbol{z}}) + \log\frac{j}{j+1}\right), \\ \alpha_2 &= \exp\left(\ell_2(\boldsymbol{z}, \widetilde{\boldsymbol{z}}) - \ell_3(\boldsymbol{z}, \widetilde{\boldsymbol{z}}) - \log(j+1)\right) \end{aligned} \tag{35}$$

then s_3 and ℓ_3 are the corresponding estimates of the combined $\boldsymbol{\epsilon}_i$ batches.

Also, note that we keep most of our operations in the log space to make the procedure numerically stable. For example, we use the $\texttt{logaddexp}(\ell_1, \ell_2)$ operation, which allows to numerically stable compute $\log(\exp(\ell_1)+\exp(\ell_2))$, and the $\texttt{logsumexp}$ trick to compute ℓ_1 and ℓ_2 themselves. Applying this algorithm iteratively allows us to process an arbitrarily large number of samples $\boldsymbol{\epsilon}_i$ while keeping the memory requirement constant. As a direct consequence, we note that our score gradient estimation is completely parallelizable.

3.3. Algorithms

Following the previous findings, we propose two new algorithms for SIVI.

Algorithm 1 BSIVI

Input: target density $p_{\boldsymbol{z}}$, batch size m, number of latent samples k with $k > m$, SIVI model $h_{\boldsymbol{\phi}}$
$i = 1, \dots, m, \quad j = 1, \dots, k$
repeat
  $\boldsymbol{\epsilon}_j \sim p_{\boldsymbol{\epsilon}}, \boldsymbol{\eta}_j \sim p_{\boldsymbol{\eta}}$
  $\boldsymbol{z}_i = h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}_i, \boldsymbol{\eta}_i)$
  $s_i = \nabla_{\boldsymbol{z}_i} \texttt{logsumexp}\left(\left\{\log q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}_i|\boldsymbol{\epsilon}_j)\right\}_{j=1,\dots,k}\right)$
  $q_i = \texttt{stop\_gradient}(s_i) \cdot \boldsymbol{z}_i$
  $\text{loss} = 1/m \sum_{i=1}^{m} (q_i - \log p_{\boldsymbol{z}}(\boldsymbol{z}_i))$
  $\boldsymbol{\phi} = \texttt{opt}(\text{loss}, \boldsymbol{\phi})$
until $\boldsymbol{\phi}$ has converged

3.3.1. BSIVI

As a new baseline method, we propose base SIVI (BSIVI), which minimizes the reverse Kullback-Leibler divergence $D_{\mathrm{KL}}(q_{\boldsymbol{z}} \| p_{\boldsymbol{z}})$ by following the path gradient of Eq. 7. For the score gradient $\nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}}$ we plug-in $s_{\mathrm{MC},k}(\boldsymbol{z})$. This method exploits the fact that we can rapidly sample from a SIVI model, and $s_{\mathrm{MC},k}$ can be computed with constant memory independent of k as discussed in Section 3.2. The algorithm is summarized in Algorithm 1. We use BSIVI to ablate the use of importance sampling, which our main method is built upon.

3.3.2. AISIVI

Furthermore, we propose adaptively informed SIVI (AISIVI), which alternates between minimizing the expected forward KL divergence $E_{\boldsymbol{z}\sim q_{\boldsymbol{z}}}\left[D_{\mathrm{KL}}(q_{\boldsymbol{\epsilon}|\boldsymbol{z}} \| \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}})\right]$ and the reverse KL divergence $D_{\mathrm{KL}}(q_{\boldsymbol{z}} \| p_{\boldsymbol{z}})$ by following the path gradient of Eq. 7. For the score gradient $\nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}}$, we plug-in $s_{\mathrm{IS},k}(\boldsymbol{z})$, which uses $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ as the proposal distribution. This alternating training is possible since $s_{\mathrm{IS},k}(\boldsymbol{z})$ is a consistent estimator of the score gradient for any $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ with $\mathrm{supp}(q_{\boldsymbol{\epsilon}|\boldsymbol{z}}) \subset \mathrm{supp}(\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}})$. Since the forward D_{KL} is mass covering, we can expect that the support assumption is always fulfilled. This means, in contrast to UIVI, we do not need exact[4] samples from $q_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ and the bias and variance of our estimate decreases[5] with increasing k. Also, sampling from the CNF $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ is comparatively cheap, and the samples are guaranteed to be independent.

4. Related Literature

Yin & Zhou (2018) propose to use semi-implicit distributions for VI and train their models by sandwiching the ELBO. Titsias & Ruiz (2019) introduce another objective based on ELBO and derive an associated unbiased gradient

[4] However, we can greatly reduce the bias the better we match $q_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ with $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$

[5] This is not the case for UIVI regarding the number of chains

Algorithm 2 AISIVI

Input: target density $p_{\boldsymbol{z}}$, batch size m, number of latent samples k, SIVI model $h_{\boldsymbol{\phi}}$, CNF $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$
$i = 1, \dots, m, \quad j = 1, \dots, k$
repeat
  $\boldsymbol{\epsilon}_i \sim p_{\boldsymbol{\epsilon}}, \boldsymbol{\eta}_i \sim p_{\boldsymbol{\eta}}$
  $\boldsymbol{z}_i = h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}_i, \boldsymbol{\eta}_i)$
  $\text{loss}_{\text{flow}} = -1/m \sum_{i=1}^{m} \log \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}_i|\boldsymbol{z}_i)$
  $\boldsymbol{\theta} = \texttt{opt}(\text{loss}_{\text{flow}}, \boldsymbol{\theta})$

  $\boldsymbol{\epsilon}_{i,j} \sim \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\cdot|\boldsymbol{z}_i)$
  $\log w_{i,j} = \log p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}_{i,j}) - \log \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}_{i,j}|\boldsymbol{z}_i)$
  $\log \widetilde{w}_{i,j} = \texttt{stop\_gradient}(\log w_{i,j})$
  $\log \widetilde{q}_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}_i|\boldsymbol{\epsilon}_{i,j}) = \log \widetilde{w}_{i,j} + \log q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}_i|\boldsymbol{\epsilon}_{i,j})$
  $s_i = \nabla_{\boldsymbol{z}_i} \texttt{logsumexp}\left(\left\{\log \widetilde{q}_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}_i|\boldsymbol{\epsilon}_{i,j})\right\}_{j=1,\dots,k}\right)$
  $q_i = \texttt{stop\_gradient}(s_i) \cdot \boldsymbol{z}_i$
  $\text{loss} = 1/m \sum_{i=1}^{m} (q_i - \log p_{\boldsymbol{z}}(\boldsymbol{z}_i))$
  $\boldsymbol{\phi} = \texttt{opt}(\text{loss}, \boldsymbol{\phi})$
until ϕ has converged

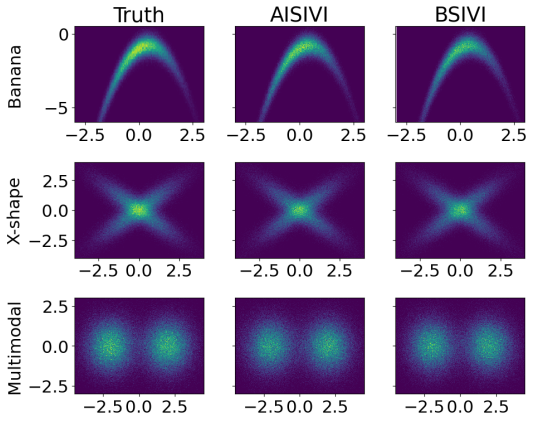

Figure 2. Histograms based on 100000 samples produced by the true distribution, AISIVI, and BSIVI

estimator, which, however, depends on expensive MCMC simulations. Sobolev & Vetrov (2019) also improved upon Yin & Zhou (2018) by introducing an importance sampling distribution; however, using expressive models such as CNFs remains infeasible for their approach. In recent years, new approaches based on different objectives have been proposed that seem to outperform methods based on the ELBO. Yu & Zhang (2023) propose minimizing the Fisher divergence, but their minimax formulation proves difficult to train compared to the standard minimization problems mentioned above.

Building upon Yu & Zhang (2023), Cheng et al. (2024) use the kernel Stein discrepancy as the training objective, which turns the minimax problem into a standard minimization problem. We will refer to their method as KSIVI. Lim & Johansen (2024) proposed Particle Semi-Implicit Variational Inference (PVI), which is a particle approximation of a Euclidean-Wasserstein gradient flow. Both Cheng et al. (2024) and Lim & Johansen (2024) showed strong empirical evidence supporting their methods.

While beyond the scope of this work, we note that SIVI has been successfully extended to multilayer architectures, yielding improved performance as demonstrated by Yu et al. (2023).

Beyond SIVI, another line of research explores variational inference with fully implicit distributions (Mescheder et al., 2017; Shi et al., 2018; Feng et al., 2017). These methods often encounter training challenges, such as instability introduced by adversarial learning or density-ratio estimation.

Another related direction performs inference directly in function space (Sun et al., 2019; Ma et al., 2019; Pielok et al., 2023). These approaches frequently incorporate implicit inference mechanisms within their frameworks.

Several approaches have improved variational inference by incorporating importance sampling. IWAE (Burda et al., 2016) introduces a tighter bound through multiple importance-weighted samples, while NVI (Zimmermann et al., 2021) extends this idea using nested objectives to learn better proposal distributions. Our work builds on this line by integrating importance sampling into the SIVI framework to improve expressivity and stability.

5. Experiments

In the following, we analyze the performance of our proposed methods AISIVI and BSIVI under different data scenarios. We start by comparing our two methods on well-known toy examples that serve as a first sanity check (Section 5.1). We then compare our methods with the state-of-the-art methods KSIVI and PVI on a 22-dimensional problem in the context of a Bayesian logistic regression model (Section 5.2, which serves as another common benchmark example for SIVI. Finally, we move to a 100-dimensional problem related to a conditioned diffusion process (Section 5.3). We implemented AISIVI and BSIVI in PyTorch (Paszke et al., 2019). All experiments are performed on a Linux-based server A5000 server with 2 GPUs, 24GB VRAM, and Intel Xeon Gold 5315Y processor with 3.20 GHz.

Table 1. $D_{\mathrm{KL}}(p, q)$ of different toy examples (rows) using the two proposed methods (columns).

NAME	↓ AISIVI (D_{KL})	↓ BSIVI (D_{KL})
BANANA	0.0853	0.3022
MULTIMODAL	0.0044	0.0017
X-SHAPE	0.0072	0.0034

5.1. Toy examples

First, we train BSIVI and AISIVI on the three common two-dimensional test densities Banana, X-Shape, and Multimodal as proposed by Cheng et al. (2024). Their respective definitions can be found in Table 3 in the Appendix B. For both methods, we use the same NN architecture and train them for 4000 iterations. For the NF of AISIVI, we use 6 conditional affine coupling layers.

Results It can be seen in Figure 2 that AISIVI and BSIVI can capture the three densities nearly equally well. Only for the Banana benchmark, AISIVI outperforms BSIVI notably (Table 1).

5.2. Bayesian Logistic Regression

Next, we perform a Bayesian logistic regression on the WAVEFORM[6] dataset as proposed by Yin & Zhou (2018). For the target variables $y_i \in \{0, 1\}, i = 1, \ldots, N$ with $N = 400$ and the feature vectors $\boldsymbol{x}_i \in \mathbb{R}^{21}$, the log-likelihood is given by

$$\log p(y_{1,\ldots,N} | \, \boldsymbol{x}_{1,\ldots,N}, \boldsymbol{\beta}) = \sum_{i=1}^{N} y_i (1, \boldsymbol{x}_i^\top)\boldsymbol{\beta} - \log\left(1 + \exp\left((1, \boldsymbol{x}_i^\top)\boldsymbol{\beta}\right)\right),$$

where $\boldsymbol{\beta} \in \mathbb{R}^{22}$ is the variable we want to infer. We set the prior distribution of $\boldsymbol{\beta}$ to a normal distribution, i.e., $p(\boldsymbol{\beta}) = \mathcal{N}(0, \alpha^{-1} I)$ with $\alpha = 0.01$. In line with Cheng et al. (2024), we estimate the ground truth by simulating parallel stochastic gradient Langevin dynamics (SGLD Welling & Teh, 2011) for 400,000 iterations, 1000 samples, and a step size of 0.0001. We use the same NN architecture for all methods and use the best hyperparameters for PVI and KSIVI proposed by the respective authors for this benchmark. We train AISIVI and BSIVI for 10,000 iterations and use $\boldsymbol{\epsilon}_i$ batch sizes of 9182 and 91,820 respectively. The large batch size of BSIVI is possible and computationally feasible because of the considerations discussed in Section 3.2. All methods use a batch size $m = 128$ the latent dimension is set to 10, i.e., $\boldsymbol{\epsilon} \in \mathbb{R}^{10}$. For the NF of AISIVI, we use 16 conditional affine coupling layers. We use the full batch for the score gradient computation of the target density.

[6] https://archive.ics.uci.edu/ml/machine-learningdatabases/waveform

Table 2. KSIVI serves as the gold standard, with AISIVI reaching it in 10K iterations. The other SIVI variants are compared based on their estimated log marginal likelihood, given a comparable computational budget to AISIVI. The log marginal likelihood is estimated using 1000 high-quality SGLD samples, while each variant's estimate is computed using 60,000 samples,

METHOD	↑ LOG ML	TRAINING TIME [S]	ITERATIONS
KSIVI	74521	0.6K	100K
AISIVI	**74062**	1.4K	10K
IWHVI	67667	1.5K	10K
BSIVI	60556	1.5K	10K
PVI	53121	1.4K	10K
UIVI	40207	1.5K	10K

Results The marginal and pairwise density estimates in Figure 3 highlight that all methods perform nearly equally well since no systematic over- or underestimation of the variance can be observed. We also compare with the ground truth all pairwise correlation coefficients of $\boldsymbol{\beta}$ given by

$$\rho_{i,j} = \frac{\operatorname{cov}\left(\boldsymbol{\beta}_{(i)}, \boldsymbol{\beta}_{(j)}\right)}{\sqrt{\operatorname{cov}\left(\boldsymbol{\beta}_{(i)}, \boldsymbol{\beta}_{(i)}\right) \operatorname{cov}\left(\boldsymbol{\beta}_{(j)}, \boldsymbol{\beta}_{(j)}\right)}}, i \neq j, \quad (36)$$

where $\boldsymbol{\beta}_{(i)}$ is a vector containing the i-th coordinate of all $\boldsymbol{\beta}$ samples.

The scatter plot in Figure 4 provides a visual summary of the correlation coefficients and the relation between those of different IVI methods and the ones of SGLD as considered ground truth. The results illustrate that PVI and KSIVI exhibit a slightly reduced spread compared to our proposed methods, indicating a marginally better fit. However, overall, the performance of all methods remains comparable.

5.3. Conditioned Diffusion Process

We adopt the Bayesian inference setting proposed in Cheng et al. (2024), which is based on the Langevin stochastic differential equation (SDE):

$$dx_t = 10 x_t (1 - x_t^2) dt + dw_t, \quad 0 \leq t \leq 1, \quad (37)$$

where $x_0 = 0$ and w_t is a one-dimensional standard Brownian motion. This SDE models the motion of a particle in an energy potential with Brownian fluctuations (Detommaso et al., 2018).

Following (Cheng et al., 2024), we discretize the SDE using the Euler-Maruyama scheme with a step size $\Delta t = 0.01$, yielding a 100-dimensional latent variable

$$\boldsymbol{x} = (x_{\Delta t}, x_{2\Delta t}, \ldots, x_{100\Delta t}),$$

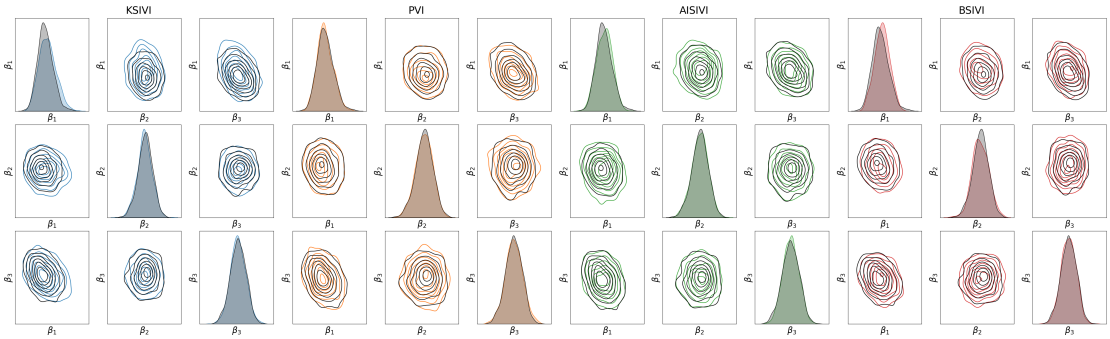

Figure 3. Comparision of marginal and pairwise density estimates of $\boldsymbol{\beta}_{(1)}, \boldsymbol{\beta}_{(2)}, \boldsymbol{\beta}_{(3)}$ where the SGLD estimates are marked in black

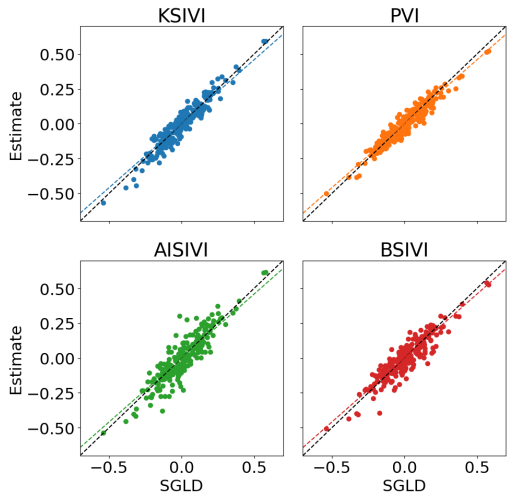

Figure 4. Scatter plot of every pairwise correlation coefficient $\rho_{i,j}$ between the estimates and SGLD.

which gives rise to the prior distribution $p_{\text{prior}}(\boldsymbol{x})$. The observations are perturbed at 20 time points, given by

$$\boldsymbol{y} = (y_{5\Delta t}, y_{10\Delta t}, \ldots, y_{100\Delta t}),$$

where

$$y_{5k\Delta t} \sim \mathcal{N}(x_{5k\Delta t}, \sigma^2), \quad 1 \leq k \leq 20 \tag{38}$$

with $\sigma = 0.1$, defining the likelihood function $p(\boldsymbol{y}|\boldsymbol{x})$. Given $\boldsymbol{y}$, our goal is to infer the posterior

$$p(\boldsymbol{x}|\boldsymbol{y}) \propto p_{\text{prior}}(\boldsymbol{x}) p(\boldsymbol{y}|\boldsymbol{x}). \tag{39}$$

To approximate the posterior, we reapply the approach in (Cheng et al., 2024) by running a long-run parallel stochastic gradient Langevin dynamics (SGLD) simulation with 1000 independent particles, a step size of 0.0001, and 100,000 iterations to generate 1000 ground truth samples.

For this benchmark, we also include IWHI to ablate the effect of their joint training approach compared to our sequential training. For their method, we use a conditional Gaussian model, where the conditional parameters are predicted by a neural network, as their joint training setup makes more complex conditional models—such as continuous normalizing flows (CNFs)—infeasible. Additionally, we evaluate against UIVI to compare our importance sampling-based enhancement with their original MCMC-based approach. For all methods, we use the same NN architecture. For KSIVI, we use the hyperparameters proposed by the authors for this benchmark. For PVI, we use 100 particles. To ensure a fair comparison, we fixed the outer batch size (number of sampled $\boldsymbol{z}$) for all SIVI methods and adjusted the inner batch size (number of sampled $\boldsymbol{\epsilon}$) until we achieved approximately the same iterations per second as AISIVI. The $\boldsymbol{\epsilon}_i$ batch sizes for AISIVI, BSIVI, and IWHI are 256, 40960, and 7000, respectively. The latent dimension is 100 for all SIVI variants. For the NF of AISIVI, we use 32 conditional affine coupling layers.

Results The results of the experiment is depicted in Figure 5. We observe that KSIVI and AISIVI are closest to SGLD while UIVI, PVI, and BSIVI tend to underestimate the variability of the process. In Table 2, we report the estimated log marginal likelihoods of the SIVI variants along with their associated training times. Notably, only our method, AISIVI, approaches the performance of the state-of-the-art KSIVI. While IWHI also performs well, it does not match AISIVI, highlighting the benefits of a more expressive proposal model. For UIVI, we were limited to an inner batch size of 2 due to computational constraints, which led to noticeably weaker performance. Nevertheless, this

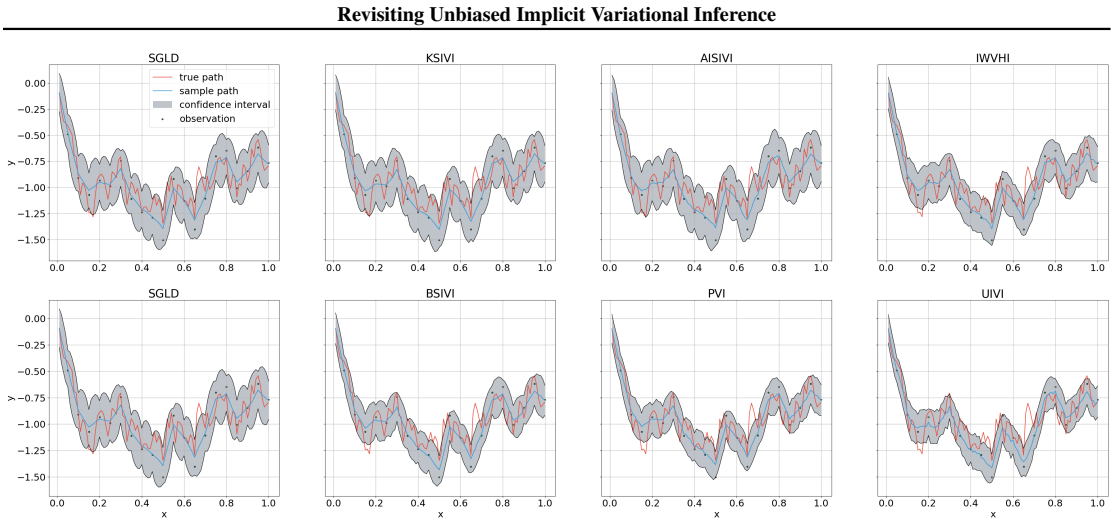

Figure 5. Approximations of KSIVI, PVI, IWHVI, UIVI, AISIVI, and BSIVI for the discretized conditioned diffusion process are shown. The red dots represent the observations, the magenta line the ground truth estimated via parallel SGLD, and the blue line the estimated posterior mean. The shaded region shows the 95 marginal posterior confidence interval at each discretization step

comparison shows that AISIVI successfully adapts UIVI's core ideas in a way that makes them more computationally efficient and competitive.

6. Conclusion

In this paper, we proposed a novel SIVI framework, AISIVI, which revitalizes the ELBO as the training objective. This is possible because the bias and variance of the ELBO gradients can be severely reduced by using importance sampling and the optimal proposal distribution can be stably learned with a CNF. We provided the respective efficient Monte Carlo gradient estimators. The numerical experiments support the efficiency and effectiveness claim of AISIVI.

In particular, our experiments on the high-dimensional diffusion example suggest that it can be beneficial not to rely on a kernel method, which is known to be scalable to very large dimensions. Our method thus represents an easy-to-use and scalable alternative to current state-of-the-art SIVI methods with on-par performance.

Limitations and Outlook

This work marks an initial attempt to integrate the strengths of semi-implicit distributions and normalizing flows. However, given the numerous normalizing flow frameworks, certain alternative combinations may lead to improved performance. Future research could explore these possibilities to identify more effective configurations. While our method shows on par performance with current state-of-the-art SIVI methods, a suitable combination could further notably enhance performance.

Additionally, the proposed method does not inherently offer exploration capabilities, which may limit its ability to model multi-modal distributions. However, note that we can always combine a temperature annealing strategy (Rezende & Mohamed, 2015) with our approach, but a more principled procedure would be desirable. While this limitation is common in related work, addressing it in future research could enhance the applicability of AISIVI.

Impact Statement

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Andrade, D. Stabilizing training of affine coupling layers for high-dimensional variational inference. *Machine Learning: Science and Technology*, 5, 12 2024. doi: 10.1088/2632-2153/ad9a39.

Burda, Y., Grosse, R. B., and Salakhutdinov, R. Importance weighted autoencoders. In Bengio, Y. and LeCun, Y. (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL

http://arxiv.org/abs/1509.00519.

Cheng, Z., Yu, L., Xie, T., Zhang, S., and Zhang, C. Kernel semi-implicit variational inference. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=w5oUo0LhO1.

Detommaso, G., Cui, T., Spantini, A., Marzouk, Y., and Scheichl, R. A stein variational newton method. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 9187–9197, Red Hook, NY, USA, 2018. Curran Associates Inc.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Feng, Y., Wang, D., and Liu, Q. Learning to draw samples with amortized stein variational gradient descent. In Elidan, G., Kersting, K., and Ihler, A. (eds.), *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017. URL http://auai.org/uai2017/proceedings/papers/206.pdf.

Figurnov, M., Mohamed, S., and Mnih, A. Implicit reparameterization gradients. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/92c8c96e4c37100777c7190b76d28233-Paper.pdf.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Lim, J. N. and Johansen, A. M. Particle semi-implicit variational inference. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=p3gMGkHMkM.

Lu, Y. and Huang, B. Structured output learning with conditional generative flows. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 5005–5012. AAAI Press, 2020. doi: 10.1609/AAAI.V34I04.5940. URL https://doi.org/10.1609/aaai.v34i04.5940.

Ma, C., Li, Y., and Hernandez-Lobato, J. M. Variational implicit processes. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4222–4233. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ma19b.html.

Mescheder, L., Nowozin, S., and Geiger, A. Adversarial variational bayes: unifying variational autoencoders and generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 2391–2400. JMLR.org, 2017.

Molchanov, D., Kharitonov, V., Sobolev, A., and Vetrov, D. Doubly semi-implicit variational inference. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 2593–2602. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/molchanov19a.html.

Owen, A. B. *Monte Carlo theory, methods and examples*. https://artowen.su.domains/mc/, 2013.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(1), January 2021. ISSN 1532-4435.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Pielok, T., Bischl, B., and Rügamer, D. Approximate bayesian inference with stein functional variational gradient descent. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=a2-aoqmeYM4.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1530–1538, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/rezende15.html.

Roeder, G., Wu, Y., and Duvenaud, D. K. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/e91068fff3d7fa1594dfdf3b4308433a-Paper.pdf.

Shi, J., Sun, S., and Zhu, J. Kernel implicit variational inference. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1l4eQW0Z.

Sobolev, A. and Vetrov, D. P. Importance weighted hierarchical variational inference. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5737c6ec2e0716f3d8a7a5c4e0de0d9a-Paper.pdf.

Sun, S., Zhang, G., Shi, J., and Grosse, R. FUNCTIONAL VARIATIONAL BAYESIAN NEURAL NETWORKS. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rkxacs0qY7.

Titsias, M. K. and Ruiz, F. Unbiased implicit variational inference. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 167–176. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/titsias19a.html.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *International Conference on Machine Learning*, 2011. URL https://api.semanticscholar.org/CorpusID:2178983.

Yin, M. and Zhou, M. Semi-implicit variational inference. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5660–5669. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/yin18b.html.

Yu, L. and Zhang, C. Semi-implicit variational inference via score matching. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sd90a2ytrt.

Yu, L., Xie, T., Zhu, Y., Yang, T., Zhang, X., and Zhang, C. Hierarchical semi-implicit variational inference with application to diffusion model acceleration. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Zimmermann, H., Wu, H., Esmaeili, B., and van de Meent, J. Nested variational inference. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 20423–20435, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ab49b208848abe14418090d95df0d590-Abstract.html.

A. Proofs

For the proofs, we assume that the objects of interest are sufficiently regular s.t. we can change the order of integration, summation, and differentiation.

A.1. $s_{\mathrm{MC},k}$ is a consistent estimator and its bias approximation

We approximate the bias of $s_{\mathrm{MC},k}(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \log\left(\frac{1}{k}\sum_{i=1}^{k} q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)\right)$ by using the delta method. First we note for large k that $s_{\mathrm{MC},k}(\boldsymbol{z}) \approx \underbrace{\nabla_{\boldsymbol{z}} \log\left(\frac{1}{k-1}\sum_{i=2}^{k} q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)\right)}_{=:\widetilde{s_{\mathrm{MC},k}}(\boldsymbol{z})}$. With the second-order Taylor approximation around $q_{\boldsymbol{z}}(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{\epsilon}_i \sim p_{\boldsymbol{\epsilon}}}\left[\frac{1}{k}\sum_{i=2}^{k-1} q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)\right]$ we get that

$$\log\left(\frac{1}{k-1}\sum_{i=2}^{k} q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)\right) \approx \log\left(q_{\boldsymbol{z}}(\boldsymbol{z})\right) + \frac{\frac{1}{k-1}\sum_{i=2}^{k} q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i) - q_{\boldsymbol{z}}(\boldsymbol{z})}{q_{\boldsymbol{z}}(\boldsymbol{z})} - \frac{\left(\frac{1}{k-1}\sum_{i=2}^{k} q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i) - q_{\boldsymbol{z}}(\boldsymbol{z})\right)^2}{2 \cdot q_{\boldsymbol{z}}(\boldsymbol{z})^2}. \tag{40}$$

From this, it follows that

$$\mathbb{E}_{\boldsymbol{\epsilon}_i \sim p_{\boldsymbol{\epsilon}}}\left[\log\left(\frac{1}{k-1}\sum_{i=2}^{k} q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)\right)\right] \approx \log\left(q_{\boldsymbol{z}}(\boldsymbol{z})\right) - \frac{\mathbb{V}_{\boldsymbol{\epsilon} \sim p_{\boldsymbol{\epsilon}}}\left[q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon})\right]}{2(k-1) \cdot q_{\boldsymbol{z}}(\boldsymbol{z})^2}. \tag{41}$$

Consequently, we get for large k that

$$\mathbb{E}_{\boldsymbol{\epsilon}_i \sim p_{\boldsymbol{\epsilon}}}\left[\nabla_{\boldsymbol{z}} s_{\mathrm{MC},k}(\boldsymbol{z})\right] - \nabla_{\boldsymbol{z}} \log\left(q_{\boldsymbol{z}}(\boldsymbol{z})\right) \approx \mathbb{E}_{\boldsymbol{\epsilon}_i \sim p_{\boldsymbol{\epsilon}}}\left[\nabla_{\boldsymbol{z}} \widetilde{s_{\mathrm{MC},k}}(\boldsymbol{z})\right] - \nabla_{\boldsymbol{z}} \log\left(q_{\boldsymbol{z}}(\boldsymbol{z})\right) \tag{42}$$

$$\approx -\nabla_{\boldsymbol{z}}\left(\frac{\mathbb{V}_{\boldsymbol{\epsilon} \sim p_{\boldsymbol{\epsilon}}}\left[q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon})\right]}{2(k-1) \cdot q_{\boldsymbol{z}}(\boldsymbol{z})^2}\right), \tag{43}$$

which, in general, is non-zero.

To prove the consistency of $s_{\mathrm{MC},k}$, we observe since $\log$ is a continuous function that

$$\lim_{k\to\infty} s_{\mathrm{MC},k} = \nabla_{\boldsymbol{z}} \log\left(\lim_{k\to\infty} \frac{1}{k}\sum_{i=1}^{k} q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)\right) \overset{\text{a.s.}}{=} \nabla_{\boldsymbol{z}} \log\left(\mathbb{E}_{\boldsymbol{\epsilon} \sim p_{\boldsymbol{\epsilon}}}\left[q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon})\right]\right) = \nabla_{\boldsymbol{z}} \log\left(q_{\boldsymbol{z}}(\boldsymbol{z})\right), \tag{44}$$

i.e.,

$$\mathbb{P}\left(\lim_{k\to\infty} s_{\mathrm{MC},k} = \nabla_{\boldsymbol{z}} \log\left(q_{\boldsymbol{z}}(\boldsymbol{z})\right)\right) = 1. \tag{45}$$

A.2. $s_{\mathrm{IS},k}$ is a consistent estimator

To prove the consistency of $s_{\mathrm{IS},k}$, we observe since $\log$ is a continuous function that

$$\lim_{k\to\infty} s_{\mathrm{IS},k} = \nabla_{\boldsymbol{z}} \log\left(\lim_{k\to\infty} \frac{1}{k}\sum_{i=1}^{k} \frac{p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}) q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)}{\tau_{\boldsymbol{\epsilon}|\widetilde{\boldsymbol{z}}}(\boldsymbol{\epsilon}_i|\widetilde{\boldsymbol{z}})}\right)\Bigg|_{\widetilde{\boldsymbol{z}}=\boldsymbol{z}} \overset{\text{a.s.}}{=} \nabla_{\boldsymbol{z}} \log\left(\mathbb{E}_{\boldsymbol{\epsilon}_i \sim \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}}\left[\frac{p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}) q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)}{\tau_{\boldsymbol{\epsilon}|\widetilde{\boldsymbol{z}}}(\boldsymbol{\epsilon}_i|\widetilde{\boldsymbol{z}})}\right]\right)\Bigg|_{\widetilde{\boldsymbol{z}}=\boldsymbol{z}}. \tag{46}$$

For a valid proposal distribution, it must hold that $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ must be non-zero where $p_{\boldsymbol{\epsilon}} \cdot q_{\boldsymbol{z}|\boldsymbol{\epsilon}} = q_{\boldsymbol{z},\boldsymbol{\epsilon}}$ is greater than zero (Owen, 2013). Consequently, the support of $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ must also contain the support of $q_{\boldsymbol{\epsilon}|\boldsymbol{z}} = \frac{q_{\boldsymbol{z},\boldsymbol{\epsilon}}}{q_{\boldsymbol{z}}}$. In this case

$$\lim_{k\to\infty} s_{\mathrm{IS},k} \overset{\text{a.s.}}{=} \nabla_{\boldsymbol{z}} \log\left(q_{\boldsymbol{z}}(\boldsymbol{z})\right), \text{ i.e., } \mathbb{P}\left(\lim_{k\to\infty} s_{\mathrm{IS},k} = \nabla_{\boldsymbol{z}} \log\left(q_{\boldsymbol{z}}(\boldsymbol{z})\right)\right) = 1. \tag{47}$$

A.3. s_3 gives the correct score gradient estimator regarding all $\boldsymbol{\epsilon}_i$ samples

Assume we got a $\boldsymbol{\epsilon}$ batch of size $(j+1)\cdot b$ and have computed the following estimators

$$s_1(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \ell_1(\boldsymbol{z}, \widetilde{\boldsymbol{z}})\Big|_{\widetilde{\boldsymbol{z}}=\boldsymbol{z}} \text{ with} \tag{48}$$

$$\ell_1(\boldsymbol{z}, \widetilde{\boldsymbol{z}}) = \log\left(\frac{1}{j\cdot b}\sum_{i=1}^{j\cdot b} w(\boldsymbol{\epsilon}_i|\widetilde{\boldsymbol{z}}) q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)\right), \tag{49}$$

$$s_2(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \ell_2(\boldsymbol{z}, \widetilde{\boldsymbol{z}})\Big|_{\widetilde{\boldsymbol{z}}=\boldsymbol{z}} \text{ with} \tag{50}$$

$$\ell_2(\boldsymbol{z}, \widetilde{\boldsymbol{z}}) = \log\left(\frac{1}{b}\sum_{i=j\cdot b+1}^{(j+1)\cdot b} w(\boldsymbol{\epsilon}_i|\widetilde{\boldsymbol{z}}) q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)\right). \tag{51}$$

These estimates can be aggregated such that

$$\ell_3(\boldsymbol{z}, \widetilde{\boldsymbol{z}}) = \operatorname{logaddexp}\left(\ell_1(\boldsymbol{z}, \widetilde{\boldsymbol{z}}) + \log j, \ell_2(\boldsymbol{z}, \widetilde{\boldsymbol{z}})\right) - \log(j+1), \tag{52}$$

$$= \log\left(\frac{1}{(j+1)\cdot b}\sum_{i=1}^{(j+1)\cdot b} w(\boldsymbol{\epsilon}_i|\widetilde{\boldsymbol{z}}) q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}|\boldsymbol{\epsilon}_i)\right), \tag{53}$$

$$s_3(\boldsymbol{z}) = \alpha_1 s_1(\boldsymbol{z}) + \alpha_2 s_2(\boldsymbol{z}) \text{ with} \tag{54}$$

$$\alpha_1 = \exp\left(\ell_1(\boldsymbol{z}, \widetilde{\boldsymbol{z}}) - \ell_3(\boldsymbol{z}, \widetilde{\boldsymbol{z}}) + \log\frac{j}{j+1}\right), \tag{55}$$

$$\alpha_2 = \exp\left(\ell_2(\boldsymbol{z}, \widetilde{\boldsymbol{z}}) - \ell_3(\boldsymbol{z}, \widetilde{\boldsymbol{z}}) - \log(j+1)\right). \tag{56}$$

$$\tag{57}$$

For the score gradient estimate, it follows that

$$s_3 = \frac{1}{\exp\left(\ell_3(\boldsymbol{z}, \widetilde{\boldsymbol{z}})\right)} \nabla_{\boldsymbol{z}} \frac{j}{j+1} \exp\left(\ell_1(\boldsymbol{z}, \widetilde{\boldsymbol{z}})\right)\Big|_{\widetilde{\boldsymbol{z}}=\boldsymbol{z}} + \frac{1}{\exp\left(\ell_3(\boldsymbol{z}, \widetilde{\boldsymbol{z}})\right)} \nabla_{\boldsymbol{z}} \frac{1}{j+1} \exp\left(\ell_2(\boldsymbol{z}, \widetilde{\boldsymbol{z}})\right)\Big|_{\widetilde{\boldsymbol{z}}=\boldsymbol{z}} \tag{58}$$

$$= \frac{1}{\exp\left(\ell_3(\boldsymbol{z}, \widetilde{\boldsymbol{z}})\right)} \nabla_{\boldsymbol{z}} \exp\left(\ell_3(\boldsymbol{z}, \widetilde{\boldsymbol{z}})\right)\Big|_{\widetilde{\boldsymbol{z}}=\boldsymbol{z}} \tag{59}$$

$$= \nabla_{\boldsymbol{z}} \ell_3(\boldsymbol{z}, \widetilde{\boldsymbol{z}})\Big|_{\widetilde{\boldsymbol{z}}=\boldsymbol{z}}. \tag{60}$$

B. Implementation Details

Table 3 summarizes the details for the toy example discussed in Section 5.1.

Table 3. Densities of the toy examples

NAME	DENSITY	PARAMETERS
BANANA	$z = (\nu_1, \nu_1^2 + \nu_2 + 1)^\top, \nu \sim \mathcal{N}(0, \Sigma)$	$\Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$
MULTIMODAL	$z \sim 0.5\mathcal{N}(z\mid \mu_1, I) + 0.5\mathcal{N}(z\mid \mu_2, I)$	$\mu_1 = (-2, 0)^\top, \mu_2 = (2, 0)^\top$
X-SHAPE	$z \sim 0.5\mathcal{N}(z\mid 0, \Sigma_1) + 0.5\mathcal{N}(z\mid 0, \Sigma_2)$	$\Sigma_1 = \begin{bmatrix} 2 & 1.8 \\ 1.8 & 2 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 2 & -1.8 \\ -1.8 & 2 \end{bmatrix}$

5. Semi-Implicit Variational Inference via Kernelized Path Gradient Descent

Contributing article

Tobias Pielok, Bernd Bischl, and David Rügamer. 2026. Semi-Implicit Variational Inference via Kernelized Path Gradient Descent. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*. https://arxiv.org/abs/2506.05088.

Copyright information

Copyright © 2026 by the authors and AISTATS.

Author contributions

Tobias Pielok was solely responsible for the development of the core ideas, theoretical derivations, and all implementations and experiments. David Rügamer contributed to the writing process through collaborative editing and refinement of the manuscript. Bernd Bischl supported the work with insightful comments on writing and benchmarking, and helped shape the broader framing of the contribution.

Semi-Implicit Variational Inference via Kernelized Path Gradient Descent

Tobias Pielok
LMU Munich, MCML

Bernd Bischl
LMU Munich, MCML

David Rügamer
LMU Munich, MCML

Abstract

Semi-implicit variational inference (SIVI) is a powerful framework for approximating complex posterior distributions, but training with the Kullback–Leibler (KL) divergence can be challenging due to high variance and bias in high-dimensional settings. While current state-of-the-art semi-implicit variational inference methods, particularly Kernel Semi-Implicit Variational Inference (KSIVI), have been shown to work in high dimensions, training remains moderately expensive. In this work, we propose a kernelized KL divergence estimator that stabilizes training through nonparametric smoothing. To further reduce the bias, we introduce an importance sampling correction. We provide a theoretical connection to the amortized version of the Stein variational gradient descent, which estimates the score gradient via Stein's identity, showing that both methods minimize the same objective, but our semi-implicit approach achieves lower gradient variance. In addition, our method's bias in function space is benign, leading to more stable and efficient optimization. Empirical results demonstrate that our method outperforms or matches state-of-the-art SIVI methods in both performance and training efficiency.

1 INTRODUCTION

Accurately approximating complex probability distributions is fundamental for tasks such as Bayesian inference and learning in energy-based models, where distributions—such as Gibbs distributions—are defined in terms of an energy function. In such settings, latent variables often have physical or semantic meaning, and capturing the correct structure and uncertainty of the posterior is critical for robust learning and decision-making.

Variational Inference (VI) is a powerful framework for approximating complex posterior distributions in probabilistic models. Traditional or explicit VI relies on simple, tractable families of distributions and typically minimizes the Kullback-Leibler (KL) divergence. While computationally efficient, this approach can lead to biased approximations when the variational family is too restrictive. In contrast, implicit VI leverages flexible distributions defined by sampling procedures without requiring a tractable density, enabling more expressive posteriors but often relying on adversarial or score-based techniques.

Semi-implicit variational inference (SIVI) strikes the balance between expressivity and tractability by defining variational distributions as mixtures with an implicit component.

Among the different SIVI methods (see Section 4 for related literature), Cheng et al. (2024) proposed a score-based method called KSIVI, which achieves state-of-the-art performance using kernelized Stein discrepancies to estimate gradients of implicit variational distributions. In contrast to most previous SIVI methods, KSIVI can be efficiently trained and scales well to high-dimensional problems, establishing it as a practical and state-of-the-art approach.

Our Contributions In this work, we propose a novel approach for performing inference with semi-implicit variational distributions by combining kernelized score estimation with pathwise gradients. Our contributions are as follows: (i) We introduce the *Kernelized Path Gradient (KPG)*, a method that leverages the reparameterization structure of semi-implicit distributions and enables efficient gradient-based optimization. (ii) We show that KPG yields a provably lower-variance estimator than amortized Stein varia-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

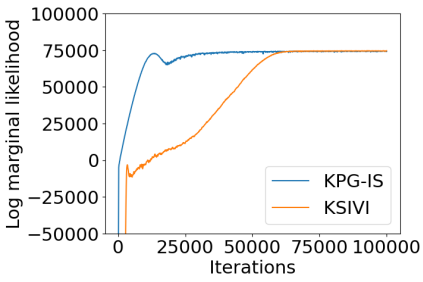

Figure 1: Convergence speed comparison between the current state-of-the-art method KSIVI and our proposal KPG-IS.

tional gradient descent (Feng et al., 2017). (iii) To further improve sample efficiency and reduce bias, we introduce *KPG-IS*, an importance-weighted variant that learns a proposal distribution for the latent variables via a constrained mixture model. (iv) We show that the optimal proposal distribution involves a tradeoff between bias and variance, and can be learned for each specific tradeoff in an unbiased manner. (v) Finally, we demonstrate empirical on-par performance or improvements over state-of-the-art semi-implicit inference methods (cf. Fig. 1).

2 BACKGROUND

2.1 Semi-implicit Variational Inference

A semi-implicit distribution $q_{\boldsymbol{z}}$ over $Z \subset \mathbb{R}^{d_Z}$ generates samples through a two-step hierarchy. First, a latent variable $\boldsymbol{\epsilon} \sim p_{\boldsymbol{\epsilon}}$ (taking values in $E \subset \mathbb{R}^{d_E}$) is passed through a neural network $f_{\boldsymbol{\phi}} : E \to Y$ ($Y \subset \mathbb{R}^{d_Y}$), whose output $\boldsymbol{y} = f_{\boldsymbol{\phi}}(\boldsymbol{\epsilon})$ parameterizes a simple explicit conditional $q_{\boldsymbol{z}|\boldsymbol{y}}$, such as a factorized Gaussian. Assuming the conditional is reparameterizable, there exists a function $g : Y \times H \to Z$ ($H \subset \mathbb{R}^{d_H}$) such that drawing $\boldsymbol{z} \sim q_{\boldsymbol{z}|\boldsymbol{y}}$ reduces to drawing $\boldsymbol{\epsilon}$ and a random variable $\boldsymbol{\eta}$ (taking values in H), i.e.,

$$\boldsymbol{z} = g(\boldsymbol{y}, \boldsymbol{\eta}) = \underbrace{g(f_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}), \boldsymbol{\eta})}_{=: h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{\eta})}. \tag{1}$$

For example, in the case where the conditional distribution is a factorized Gaussian, sampling is performed by drawing $\boldsymbol{\eta} \sim \mathcal{N}(0, I)$ and computing

$$\boldsymbol{z} = \boldsymbol{\mu}_{\boldsymbol{\epsilon}} + \mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}})\boldsymbol{\eta}, \tag{2}$$

where $\boldsymbol{\mu}_{\boldsymbol{\epsilon}}$ and $\boldsymbol{\sigma}_{\boldsymbol{\epsilon}}$ denote the mean and standard deviation vectors, respectively, which in this case are the outputs of $f_{\boldsymbol{\phi}}$. This construction enables us to express the likelihood of a sample $\boldsymbol{z}$ in a principled manner, such that

$$q_{\boldsymbol{z}}(\boldsymbol{z}) = \mathop{\mathbb{E}}_{\boldsymbol{\epsilon} \sim p_{\boldsymbol{\epsilon}}} \left[q_{\boldsymbol{z}|\boldsymbol{y}}\left(\boldsymbol{z} | f_{\boldsymbol{\phi}}(\boldsymbol{\epsilon})\right) \right] = \mathop{\mathbb{E}}_{\boldsymbol{\epsilon} \sim p_{\boldsymbol{\epsilon}}} \left[q_{\boldsymbol{z}|\boldsymbol{\epsilon}}\left(\boldsymbol{z} | \boldsymbol{\epsilon}\right) \right]. \tag{3}$$

Note also that expectations with respect to $q_{\boldsymbol{z}}$ are compatible with the reparameterization trick (Kingma and Welling, 2014); that is, for a differentiable function $\ell : Z \to \mathbb{R}$ which could possibly also depend on $\boldsymbol{\phi}$, it holds that

$$\nabla_{\boldsymbol{\phi}} \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{z}}} \left[\ell(\boldsymbol{z}) \right] = \mathbb{E}_{\boldsymbol{\epsilon}, \boldsymbol{\eta} \sim p_{\boldsymbol{\epsilon}, \boldsymbol{\eta}}} \nabla_{\boldsymbol{\phi}} \left[\ell(h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{\eta})) \right]. \tag{4}$$

2.2 Amortized Stein Variational Gradient Descent

Several amortized versions of *Stein Variational Gradient Descent* (SVGD) have been introduced in Feng et al. (2017) to enable inference through neural networks. Here, we briefly describe the most widely used formulation, which views amortized SVGD as minimizing the kernel-smoothed difference between score functions. More specifically, suppose we have a neural sampler $q_{\boldsymbol{z}}$, which means that we can generate samples by transforming noise $\boldsymbol{\xi} \sim p_{\boldsymbol{\xi}}$ through a neural network $h_{\boldsymbol{\phi}}$, but cannot evaluate the likelihood of the samples. The objective is to minimize the reverse KL divergence between $q_{\boldsymbol{z}}$ and the target distribution $p_{\boldsymbol{z}}$, i.e.,

$$D_{\mathrm{KL}}(q_{\boldsymbol{z}} \| p_{\boldsymbol{z}}) = \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{z}}} \left[\log \left(\frac{q_{\boldsymbol{z}}(\boldsymbol{z})}{p_{\boldsymbol{z}}(\boldsymbol{z})} \right) \right]. \tag{5}$$

Since $q_{\boldsymbol{z}}$ is amenable to the reparameterization trick, a low-variance gradient estimator of the KL divergence can be derived (Roeder et al., 2017). Hence, leveraging the fact that the expected score function vanishes, we obtain the *pathwise gradient estimator*, i.e.,

$$\nabla_{\boldsymbol{\phi}} D_{\mathrm{KL}}(q_{\boldsymbol{z}} \| p_{\boldsymbol{z}}) = \mathbb{E}_{\boldsymbol{\xi} \sim p_{\boldsymbol{\xi}}} \left[\Delta(h_{\boldsymbol{\phi}}(\boldsymbol{\xi})) \cdot \nabla_{\boldsymbol{\phi}} h_{\boldsymbol{\phi}}(\boldsymbol{\xi}) \right]. \tag{6}$$

where the difference in score gradients $\Delta(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}}(\boldsymbol{z}) - \nabla_{\boldsymbol{z}} \log p_{\boldsymbol{z}}(\boldsymbol{z})$. However, we do not have access to the score gradient $\nabla_{\boldsymbol{z}} \log q_{\boldsymbol{z}}(\boldsymbol{z})$. To address this, we first rewrite the score gradient difference such that

$$\Delta(\boldsymbol{z}) = \arg\max_{\omega \in \mathcal{H}} \mathop{\mathbb{E}}_{\boldsymbol{z} \sim q_{\boldsymbol{z}}} \left[2\omega(\boldsymbol{z})^\top \Delta(\boldsymbol{z}) - \|\omega(\boldsymbol{z})\|^2_{\mathcal{H}} \right], \tag{7}$$

where the function space $\mathcal{H}$ is the space of square integrable functions L^2. To apply Stein's identity for estimating the score gradient (Li and Turner, 2018; Liu and Wang, 2016), we utilize the result proven in Cheng et al. (2024), which states that when the function space $\mathcal{H}$ of the maximization problem defined in Eq. 7 is restricted to a reproducing kernel Hilbert space (RKHS)

with a given kernel function $k : Z \times Z \to \mathbb{R}_{\geq 0}$, the problem admits a unique solution, given explicitly by

$$\Delta_k(\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{z}' \sim q_{\boldsymbol{z}}} k(\boldsymbol{z}, \boldsymbol{z}')[\nabla_{\boldsymbol{z}'} \log q_{\boldsymbol{z}}(\boldsymbol{z}') - \nabla_{\boldsymbol{z}'} \log p_{\boldsymbol{z}}(\boldsymbol{z}')]. \tag{8}$$

Under the assumptions that the kernel k is continuously differentiable and

$$k(\boldsymbol{z}, \boldsymbol{z}') q_{\boldsymbol{z}}(\boldsymbol{z}')|_{\partial Z} = 0 \tag{9}$$

$$\text{or } \lim_{\boldsymbol{z}' \to \infty} k(\boldsymbol{z}, \boldsymbol{z}') q_{\boldsymbol{z}}(\boldsymbol{z}') = 0 \text{ if } Z = \mathbb{R}^{d_Z}, \tag{10}$$

we get that

$$\underbrace{\mathop{\mathbb{E}}_{\boldsymbol{z}' \sim q_{\boldsymbol{z}}} [-\nabla_{\boldsymbol{z}'} k(\boldsymbol{z}, \boldsymbol{z}') - k(\boldsymbol{z}, \boldsymbol{z}') \nabla_{\boldsymbol{z}'} \log p_{\boldsymbol{z}}(\boldsymbol{z}')]}_{:= \Delta_{\text{STEIN},k}(\boldsymbol{z})} = \Delta_k(\boldsymbol{z}) \tag{11}$$

by using Stein's identity, which can be proved via integration by parts. Replacing the difference in score gradients Δ in the pathwise gradient estimator given by Eq. 6 with a Monte-Carlo estimate of the kernelized difference in score gradients Δ_k in Eq. 11 results in the amortized SVGD update step.

3 METHOD

We note that the same derivation as amortized SVGD can be followed; however, rather than relying on a simple neural sampler, we leverage a semi-implicit distribution since we can just substitute $\boldsymbol{\xi}$ with $(\boldsymbol{\epsilon}, \boldsymbol{\eta})$. While we cannot directly plug Eq. 8 into Eq. 6, we can first apply the kernel trick for semi-implicit distributions, as introduced by Cheng et al. (2024), which states that

$$\mathop{\mathbb{E}}_{\boldsymbol{z}' \sim q_{\boldsymbol{z}}} [k(\boldsymbol{z}, \boldsymbol{z}') \nabla_{\boldsymbol{z}'} \log q_{\boldsymbol{z}}(\boldsymbol{z}')] = \mathbb{E}_{\boldsymbol{z}', \boldsymbol{\epsilon}' \sim q_{\boldsymbol{z},\boldsymbol{\epsilon}}} \left[k(\boldsymbol{z}, \boldsymbol{z}') \nabla_{\boldsymbol{z}'} \log q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}' | \boldsymbol{\epsilon}')\right]. \tag{12}$$

This enables us to compute a Monte Carlo estimate of the resulting expression, i.e.,

$$\Delta_{\text{SI},k}(\boldsymbol{z}) := \mathbb{E}_{\boldsymbol{z}', \boldsymbol{\epsilon}' \sim q_{\boldsymbol{z},\boldsymbol{\epsilon}}} k(\boldsymbol{z}, \boldsymbol{z}') \cdot \left[\nabla_{\boldsymbol{z}'} \log q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}' | \boldsymbol{\epsilon}') - \nabla_{\boldsymbol{z}'} \log p_{\boldsymbol{z}}(\boldsymbol{z}')\right] = \Delta_k(\boldsymbol{z}), \tag{13}$$

which can then be substituted into Eq. 6. With this, we define the Kernelized Path Gradient (KPG) as

$$\mathbb{E}_{\boldsymbol{\epsilon}, \boldsymbol{\eta} \sim p_{\boldsymbol{\epsilon},\boldsymbol{\eta}}} \left[\Delta_{\text{SI},k}(h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{\eta})) \cdot \nabla_{\boldsymbol{\phi}} h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \boldsymbol{\eta})\right]. \tag{14}$$

Exploiting the hierarchical structure of the semi-implicit distribution is a crucial distinction, as it reduces the variance of the score gradient estimator, avoids boundary assumptions, and requires only a continuous kernel. To analyze the difference in variability between the Stein gradient estimator

$$s_{\text{STEIN},k}(\boldsymbol{z}) := \mathbb{E}_{\boldsymbol{z}' \sim q_{\boldsymbol{z}}} \left[-\nabla_{\boldsymbol{z}'} k(\boldsymbol{z}, \boldsymbol{z}')\right] \tag{15}$$

and the semi-implicit gradient estimator

$$s_{\text{SI},k}(\boldsymbol{z}) := \mathop{\mathbb{E}}_{\boldsymbol{z}', \boldsymbol{\epsilon}' \sim q_{\boldsymbol{z},\boldsymbol{\epsilon}}} \left[k(\boldsymbol{z}, \boldsymbol{z}') \nabla_{\boldsymbol{z}'} \log q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}' \mid \boldsymbol{\epsilon}')\right], \tag{16}$$

we consider the trace of the difference of the covariance matrices

$$\Delta\mathbb{V} := \text{trace}\left(\mathbb{V}\left[\hat{s}_{\text{STEIN},k}(\boldsymbol{z})\right] - \mathbb{V}\left[\hat{s}_{\text{SI},k}(\boldsymbol{z})\right]\right) \tag{17}$$

of their corresponding Monte Carlo estimators $\hat{s}_{\text{STEIN},k}(\boldsymbol{z})$ and $\hat{s}_{\text{SI},k}(\boldsymbol{z})$, computed using n i.i.d. samples. Since both estimators share the same expectation, i.e., $\mathbb{E}\left[\hat{s}_{\text{STEIN},k}(\boldsymbol{z})\right] = \mathbb{E}\left[\hat{s}_{\text{SI},k}(\boldsymbol{z})\right]$, we show in Appendix A.1 that

$$\Delta\mathbb{V} = \mathbb{E}\left\|\hat{s}_{\text{STEIN},k}(\boldsymbol{z})\right\|_2^2 - \mathbb{E}\left\|\hat{s}_{\text{SI},k}(\boldsymbol{z})\right\|_2^2 \tag{18}$$

and establish the following proposition.

Proposition 3.1 *Assuming that the kernel k is the Gaussian density kernel, i.e., $k(\boldsymbol{z}, \boldsymbol{z}') = \frac{1}{(2\pi\sigma_k^2)^{d_Z/2}} \exp\left(-\frac{\|\boldsymbol{z}-\boldsymbol{z}'\|_2^2}{2\sigma_k^2}\right)$ and $q_{\boldsymbol{z}|\boldsymbol{\epsilon}}$ is a conditional Gaussian distribution, it holds for $\boldsymbol{z} \in Z$ that*

$$\mathbb{E}\left\|\hat{s}_{STEIN,k}(\boldsymbol{z})\right\|_2^2 - \mathbb{E}\left\|\hat{s}_{SI,k}(\boldsymbol{z})\right\|_2^2 = \frac{1}{n}\mathbb{E}_{\boldsymbol{\epsilon}', \boldsymbol{\eta}' \sim p_{\boldsymbol{\epsilon},\boldsymbol{\eta}}} k(\boldsymbol{z}, \text{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}' + \boldsymbol{\mu}_{\boldsymbol{\epsilon}'})^2 \cdot \left[\frac{\|\text{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}' + \boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2}{\sigma_k^4} - \|\text{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^{-1}\boldsymbol{\eta}'\|_2^2\right]. \tag{19}$$

To better understand how the difference in the variability of the score gradients depends on σ_k and $\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'}$, we first prove in Appendix A.2 a sufficient condition under which $\mathbb{E}\left\|\hat{s}_{\text{STEIN},k}(\boldsymbol{z})\right\|_2^2 \geq \mathbb{E}\left\|\hat{s}_{\text{SI},k}(\boldsymbol{z})\right\|_2^2$, providing insight into the scaling behavior with respect to these parameters.

Proposition 3.2 *Under the assumptions of Proposition 3.1, it holds that $\mathbb{E}\left\|\hat{s}_{STEIN,k}(\boldsymbol{z})\right\|_2^2 \geq \mathbb{E}\left\|\hat{s}_{SI,k}(\boldsymbol{z})\right\|_2^2$*

$$\text{if } \min(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^4 - 2\min(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^2 \max(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'}) \frac{\|\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2}{\|\boldsymbol{\eta}'\|_2} + \min(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^2 \frac{\|\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2}{\|\boldsymbol{\eta}'\|_2^2} \geq \sigma_k^4 \quad a.s. \tag{20}$$

Assuming that the maximum value of the random vector $\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'}$ is less than one[1] a.s., we observe that σ_k scales benignly with $\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'}$. Specifically, for sufficiently small maximum values of $\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'}$, the term involving $\min(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^2$ is expected to dominate the inequality since, in general, $\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} \neq \boldsymbol{z}$, suggesting that σ_k only needs to scale quadratically a.s. with $\min(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})$. This enables us to find an upper bound for the difference in the variance of the score gradient norms, with the proof provided in Appendix A.3.

Proposition 3.3 *Under the assumptions of Proposition 3.2 and additionally assuming that* $k(\boldsymbol{z}, \boldsymbol{z}')^2$ *and* $\left[\frac{\|\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}' + \boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2}{\sigma_k^4} - \|\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^{-1}\boldsymbol{\eta}'\|_2^2\right]$ *are negatively correlated, it holds that*

$$\begin{aligned}
&\mathbb{E}\,\|\hat{s}_{STEIN,k}(\boldsymbol{z})\|_2^2 - \mathbb{E}\,\|\hat{s}_{SI,k}(\boldsymbol{z})\|_2^2 \leq \\
&\frac{1}{n}\mathbb{E}_{\boldsymbol{\epsilon}',\boldsymbol{\eta}'\sim p_{\epsilon,\eta}} k(\boldsymbol{z}, \operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}' + \boldsymbol{\mu}_{\boldsymbol{\epsilon}'})^2 \cdot \\
&\left(\frac{\mathbb{E}_{\boldsymbol{\epsilon}'\sim p_\epsilon}\left[\|\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'}\|_2^2 + \|\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2\right]}{\sigma_k^4} - \mathbb{E}_{\boldsymbol{\epsilon}'\sim p_\epsilon}\left[\|\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'}^{\odot -1}\|_2^2\right]\right) \\
&=: \gamma \\
&\approx \frac{q_{\boldsymbol{z}}(\boldsymbol{z})}{n(2\sqrt{\pi}\sigma_k)^d} \cdot \gamma
\end{aligned} \tag{21}$$

where $(\cdot)^{\odot}$ *denotes the element-wise power.*

The upper bound becomes tight—that is, the inequality turns into an equality—when the correlation is zero. The assumption that $k(\boldsymbol{z}, \boldsymbol{z}')^2$ and $\left[\frac{\|\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}' + \boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2}{\sigma_k^4} - \|\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^{-1}\boldsymbol{\eta}'\|_2^2\right]$ are negatively correlated is plausible, since the first term is monotonically decreasing in $\|\boldsymbol{z} - \boldsymbol{z}'\|_2^2$, while the later term is monotonically increasing in $\|\boldsymbol{z} - \boldsymbol{z}'\|_2^2$. As $\max(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})$ decreases, the correlation between the two terms weakens, since the first term becomes asymptotically independent of $\boldsymbol{\eta}$, while the second term becomes increasingly dominated by it. Therefore, we expect the upper bound to be quite sharp in practice. Finally, we arrive at the approximate upper bound by noting that $(2\sqrt{\pi}\sigma_k)^d \cdot k^2$ is also a normalized kernel, for which the corresponding expectation converges to $q_{\boldsymbol{z}}(\boldsymbol{z})$ when σ_k is sufficiently small. From this, two key insights emerge: first, a small $\max(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})$ necessitates a correspondingly small σ_k; second, we expect that σ_k only needs to scale quadratically with $\max(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})$. As a result, the inequality becomes less restrictive due to the partial control we have over σ_k.

Also, note this flexibility is beneficial because a small σ_k is desirable—it leads to a more expressive RKHS $\mathcal{H}$, thereby reducing the bias introduced by the restriction to that RKHS. Although this might suggest using a minimal kernel width, doing so naively results in a high-variance estimator, as only a few nearby samples contribute significantly to the estimate. Moreover, in high-dimensional settings, the curse of dimensionality further exacerbates this issue, since the number of samples required to populate a local neighborhood adequately grows exponentially with the dimension, making even a large number of samples potentially insufficient.

[1] This is a trivial assumption since we can always control the upper limit of $\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'}$

3.1 Reducing the Bias via Importance Sampling

In light of these considerations, importance sampling can reduce both variance and bias. The bias-reduction mechanism is indirect: by concentrating $\boldsymbol{\epsilon}$-samples in regions of high posterior probability near $\boldsymbol{z}$, the proposal $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ causes the median heuristic to select a smaller kernel width σ_k. As noted above, a smaller σ_k yields a more expressive RKHS, directly reducing the smoothing bias, while the importance weights correct for the distributional shift from $p_{\boldsymbol{\epsilon}}$. Yet, importance sampling cannot be employed directly on $\boldsymbol{z}$, since the likelihood $q_{\boldsymbol{z}}$ of a semi-implicit distribution is intractable. However, note that we can write Δ_k such that

$$\begin{aligned}
&\Delta_{\text{SI-IS},k}(\boldsymbol{z}) = \\
&\mathbb{E}_{\boldsymbol{\epsilon}\sim\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}}\mathbb{E}_{\boldsymbol{z}'\sim q_{\boldsymbol{z}|\boldsymbol{\epsilon}}} \frac{p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})k(\boldsymbol{z}, \boldsymbol{z}')}{\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z})}\left[\nabla_{\boldsymbol{z}} \log \frac{q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}'|\boldsymbol{\epsilon})}{p_{\boldsymbol{z}}(\boldsymbol{z}')}\right] \\
&= \Delta_k(\boldsymbol{z}),
\end{aligned} \tag{22}$$

where $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ is a conditional explicit distribution with $\operatorname{supp}\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}} \supset \operatorname{supp} p_{\boldsymbol{\epsilon}}$. This means that while the direct application of importance sampling is not feasible, it can still be applied to the latent variable $\boldsymbol{\epsilon}$. This, in turn, raises the question of how to choose an optimal proposal distribution for the latent variable. Although there is no single correct choice, we adopt the following definition of optimality as it reflects a trade-off between reducing the bias by increasing density near the sample $\boldsymbol{z}$—and controlling global importance sampling variance by limiting deviation from the latent distribution. The loss is naturally induced by the likelihood and remains fully differentiable.

Definition 3.4 (Objective) *For a mixture coefficient* $\alpha(\boldsymbol{z}) \in (0, 1)$*, we define the class of proposal distributions*

$$\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z}) = \alpha(\boldsymbol{z})\, p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}) + (1 - \alpha(\boldsymbol{z}))\, \widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z}), \tag{23}$$

where $\widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ *is a learnable distribution. The correspond-*

ing optimization objective is

$$\widetilde{\tau}^*_{\boldsymbol{\epsilon}|\boldsymbol{z}} \in \arg\min_{\widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}}} \mathbb{E}_{\boldsymbol{z},\boldsymbol{\epsilon}\sim q_{\boldsymbol{z},\boldsymbol{\epsilon}}}\Big[-\log\big(\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z})q_{\boldsymbol{z}}(\boldsymbol{z})\big)\Big]. \tag{24}$$

We refer to any minimizer $\widetilde{\tau}^*_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ *of* (24) *as an* optimal proposal.

This formulation encourages proposals that interpolate between being likely under the latent prior and being adapted to samples $\boldsymbol{z} \sim q_{\boldsymbol{z}}$.

We show in Appendix A.4 the following proposition.

Proposition 3.5 *The reverse conditional distribution* $q_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ *gives the strict lower bound of our objective, i.e., the optimal distribution when* $\alpha(\boldsymbol{z})$ *converges to zero, and* $p_{\boldsymbol{\epsilon}}$ *gives the trivial strict upper bound of our objective as* $\alpha(\boldsymbol{z})$ *converges to one.*

This highlights the motivation to use the hard constraint via mixture parametrization since only

$$\operatorname{supp} q_{\boldsymbol{\epsilon}|\boldsymbol{z}} = \operatorname{supp} p_{\boldsymbol{\epsilon}} \underbrace{\frac{q_{\boldsymbol{z}|\boldsymbol{\epsilon}}}{q_{\boldsymbol{z}}}}_{\geq 0} \subset \operatorname{supp} p_{\boldsymbol{\epsilon}} \tag{25}$$

holds in general, while we require $\operatorname{supp} \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}} \supset \operatorname{supp} p_{\boldsymbol{\epsilon}}$ which is guaranteed for any $\alpha(\boldsymbol{z}) \in (0, 1]$. Therefore, a small $\alpha(\boldsymbol{z})$ is possible, but it increases the importance weight upper bound to

$$\underbrace{\lim_{\widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z})\to 0} \frac{p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})}{\alpha(\boldsymbol{z})p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}) + (1-\alpha(\boldsymbol{z}))\widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z})}}_{=1/\alpha(\boldsymbol{z})} \leq \frac{1}{\underline{\alpha}} \tag{26}$$

where $\underline{\alpha} = \inf_{\boldsymbol{z}\in Z} \alpha(\boldsymbol{z})$ likely impacting the bias-variance tradeoff. If $\widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ is arbitrarily flexible, one optimal solution for $\alpha(\boldsymbol{z})$ under our loss is always zero. However, when $\widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ lacks sufficient expressiveness, choosing $\alpha(\boldsymbol{z}) > 0$ can result in a solution closer to the optimum. Therefore, we choose to learn $\alpha(\boldsymbol{z})$ while enforcing the constraint

$$\alpha(\boldsymbol{z}) = \underline{\alpha} + (1-\underline{\alpha}) \cdot \varsigma(\widetilde{\alpha}(\boldsymbol{z})) \in (\underline{\alpha}, 1), \tag{27}$$

where $\underline{\alpha} \in (0, 1)$, ς is the sigmoid function, and $\widetilde{\alpha} : Z \to \mathbb{R}$ is an unbounded function. This parametrization enables improved approximation while maintaining stability.

Although the marginal $q_{\boldsymbol{z}}$ is inaccessible, the gradient of the objective given by Eq. 24 with respect to the joined parameters $\boldsymbol{\theta}$ of $\widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ and $\widetilde{\alpha}$ can be expressed as

$$-\mathop{\mathbb{E}}_{\boldsymbol{z},\boldsymbol{\epsilon}\sim q_{\boldsymbol{z},\boldsymbol{\epsilon}}}[\nabla_{\boldsymbol{\theta}} \log(\alpha(\boldsymbol{z})\, p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}) + (1-\alpha(\boldsymbol{z}))\, \widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z}))], \tag{28}$$

which can be estimated via Monte Carlo without bias.

Algorithm 1 KPG

Input: target density $p_{\boldsymbol{z}}$, kernel function k, batch size m, SIVI model $h_{\boldsymbol{\phi}}$,
$i = 1, \ldots, m, \quad l = 1, 2$
repeat
  $\boldsymbol{\epsilon}_{i,l} \sim p_{\boldsymbol{\epsilon}}, \boldsymbol{\eta}_{i,l} \sim p_{\boldsymbol{\eta}}$
  $\boldsymbol{z}_{i,l} = h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}_{i,l}, \boldsymbol{\eta}_{i,l})$
  $\widetilde{\boldsymbol{z}}_{i,l} = \texttt{stop\_gradient}(\boldsymbol{z}_{i,l})$
  $\widetilde{\Delta}_i(\widetilde{\boldsymbol{z}}_{i,2}) = \nabla_{\widetilde{\boldsymbol{z}}_{i,2}} \log q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\widetilde{\boldsymbol{z}}_{i,2}|\boldsymbol{\epsilon}_{i,2}) - \nabla_{\widetilde{\boldsymbol{z}}_{i,2}} \log\log p_{\boldsymbol{z}}(\widetilde{\boldsymbol{z}}_{i,2})$
  $\text{loss} = 1/m^2 \sum_{j=1}^{m} (\sum_{i=1}^{m} k(\widetilde{\boldsymbol{z}}_{j,1}, \widetilde{\boldsymbol{z}}_{i,2}) \cdot \widetilde{\Delta}_i(\widetilde{\boldsymbol{z}}_{i,2}))^\top \boldsymbol{z}_{j,1}$
  $\boldsymbol{\phi} = \texttt{opt}(\text{loss}, \boldsymbol{\phi})$
until $\boldsymbol{\phi}$ has converged

3.2 Algorithms

The algorithms developed in this work are introduced below. For both methods, the kernel width is determined in each iteration using the median heuristic (Liu and Wang, 2016).

3.2.1 KPG

We derive a straightforward Monte Carlo estimator from the KPG given by Eq. 14. The corresponding procedure is detailed in Algorithm 1. This baseline is used to ablate the effect of importance sampling, which is a key component of our main method.

3.2.2 KPG-IS

Furthermore, we propose KPG-IS, a variant of the kernel path gradient (KPG) method enhanced via importance sampling. KPG-IS alternates between minimizing the expected forward KL divergence, $E_{\boldsymbol{z}\sim q_{\boldsymbol{z}}}\left[D_{\mathrm{KL}}(q_{\boldsymbol{\epsilon}|\boldsymbol{z}} \| \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}})\right]$, and the kernelized reverse KL divergence by following the KPG defined in Eq. 14. To estimate the kernelized score gradient difference $\Delta_k(\boldsymbol{z})$, we use the importance-weighted estimator $\Delta_{\text{SI-IS},k}(\boldsymbol{z})$, which treats $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ as the proposal distribution. This alternating optimization is enabled by the fact that $s_{\text{IS},k}(\boldsymbol{z})$ is a consistent estimator of the score gradient whenever $\operatorname{supp}(p_{\boldsymbol{\epsilon}}) \supset \operatorname{supp}(\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}})$. This support condition is guaranteed by our mixture parametrization, ensuring that the estimator remains valid throughout the training.

4 RELATED LITERATURE

Yin and Zhou (2018) introduced semi-implicit variational inference (SIVI), training models by sandwiching the ELBO between upper and lower bounds. Titsias and Ruiz (2019) later proposed a related ELBO-based objective with an unbiased gradient estimator,

Algorithm 2 KPG-IS

Input: target density $p_{\boldsymbol{z}}$, kernel function k, batch size m
number of latent samples l, SIVI model $h_{\boldsymbol{\phi}}$,
conditional latent model $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$,
mixture coefficient α,
$i = 1, \dots, m, \quad j = 1, \dots, l$
repeat
$\boldsymbol{\epsilon}_i \sim p_{\boldsymbol{\epsilon}}, \boldsymbol{\eta}_i \sim p_{\boldsymbol{\eta}}$
$\boldsymbol{z}_i = h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}_i, \boldsymbol{\eta}_i)$
$\text{loss}_{\text{proposal}} = -1/m \sum_{i=1}^{m} \log \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}_i|\boldsymbol{z}_i, \alpha(\boldsymbol{z}_i))$
$\boldsymbol{\theta} = \texttt{opt}(\text{loss}_{\text{proposal}}, \boldsymbol{\theta})$

$\boldsymbol{\epsilon}_{i,j} \sim \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\cdot|\boldsymbol{z}_i), \boldsymbol{\eta}_{i,j} \sim p_{\boldsymbol{\eta}}$
$\widetilde{\boldsymbol{\zeta}}_{i,j} = \texttt{stop\_gradient}(h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}_{i,j}, \boldsymbol{\eta}_{i,j}))$
$\widetilde{\boldsymbol{z}}_i = \texttt{stop\_gradient}(\boldsymbol{z}_i)$
$\log \widetilde{w}_{i,j} = \log k(\widetilde{\boldsymbol{z}}_i, \widetilde{\boldsymbol{\zeta}}_{i,j}) + \log p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}_{i,j})$
$\quad - \log \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}_{i,j}|\widetilde{\boldsymbol{z}}_i, \alpha(\widetilde{\boldsymbol{z}}_i))$
$\widetilde{\Delta}_{i,j}(\widetilde{\boldsymbol{\zeta}}_{i,j}) = \nabla_{\widetilde{\boldsymbol{\zeta}}_{i,j}} \log q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\widetilde{\boldsymbol{\zeta}}_{i,j}|\boldsymbol{\epsilon}_{i,j}) -$
$\quad \nabla_{\widetilde{\boldsymbol{\zeta}}_{i,j}} \log p_{\boldsymbol{z}}(\widetilde{\boldsymbol{\zeta}}_{i,j})$
$\text{loss} = 1/(m \cdot l) \sum_{i=1}^{m} (\sum_{j=1}^{l} \exp(\log \widetilde{w}_{i,j}) \cdot$
$\quad \widetilde{\Delta}_i(\widetilde{\boldsymbol{\zeta}}_{i,j}))^\top \boldsymbol{z}_i$
$\boldsymbol{\phi} = \texttt{opt}(\text{loss}, \boldsymbol{\phi})$
until $\boldsymbol{\phi}$ has converged

though it requires computationally intensive MCMC sampling. Sobolev and Vetrov (2019) advanced this direction by incorporating importance sampling into the SIVI framework.

Amortized Stein Variational Gradient Descent (Feng et al., 2017) minimizes the same objective as semi-implicit methods but uses the Stein identity to compute the score gradient term. However, it does not explicitly leverage the semi-implicit structure, which can lead to higher variance in the gradient estimates due to the absence of such structural constraints.

Lim and Johansen (2024) introduced Particle Semi-Implicit Variational Inference (PVI), which approximates Euclidean-Wasserstein gradient flows using a particle-based approach and has shown promising empirical results. Meanwhile, Yu and Zhang (2023) proposed an alternative to ELBO-based training by minimizing the Fisher divergence. However, their minimax formulation introduces significant optimization challenges.

KSIVI Building on this idea, Cheng et al. (2024) replaced the Fisher divergence with the kernel Stein discrepancy, transforming the minimax objective into a standard minimization problem. We refer to this method as KSIVI, which currently stands out as the gold standard among semi-implicit variational inference methods, achieving state-of-the-art performance while maintaining computational efficiency. In our notation, the corresponding objective is

$$\text{KSD}^2(q_\phi \,\|\, p) = \mathbb{E}_{q_\phi(x,z)\, q_\phi(x',z')} \Big[k(x, x') \big\langle s_p(x) - s_{q_\phi(\cdot|z)}(x),\, s_p(x') - s_{q_\phi(\cdot|z')}(x') \big\rangle \Big]. \quad (29)$$

where $s_p(x) = \nabla_x \log p(x)$ and $s_{q_\phi(\cdot|z)}(x) = \nabla_x \log q_\phi(x \mid z)$. KSIVI minimizes this Stein discrepancy by applying the kernelized score estimator twice, leading to quadratic computational cost. In contrast, our kernelized path gradient (KPG) employs the estimator only once, resulting in substantially lower cost. Moreover, KSIVI differentiates its objective directly with respect to the variational parameters, requiring backpropagation through the score estimator, while KPG differentiates only with respect to the sample and then applies the chain rule. This methodological distinction yields a notable speed-up in practice, as also reflected in our empirical results.

5 EXPERIMENTS

We now turn to the empirical evaluation of our method. Following recent work in SIVI, the first problem is a common benchmark to test efficacy in high dimensions based on a diffusion process and initially analyzed in Cheng et al. (2024). Our second experiment tackles a Bayesian linear regression model proposed by Yin and Zhou (2018). Further common SIVI benchmarks can be found in the Appendix B. For all experiments involving KPG-IS, the proposal model $\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ is modeled as a Gaussian with a diagonal covariance structure, where the conditional parameters are learned using a neural network. As comparison methods, we use PVI (Lim and Johansen, 2024), as well as KSIVI (Cheng et al., 2024), which together form the current state-of-the-art in SIVI methods. For a fair comparison, all SIVI methods use the same neural network architecture; implementation details are provided in Appendix D. We implemented KPG and KPG-IS in PyTorch (Paszke et al., 2019). All experiments are performed on a Linux-based server A5000 server with 2 GPUs, 24GB VRAM, and Intel Xeon Gold 5315Y processor with 3.20 GHz.

5.1 Conditional Diffusion Process

To evaluate performance in high-dimensional settings, we consider a conditional diffusion process benchmark introduced in Cheng et al. (2024), which has been used to assess the effectiveness of SIVI methods. It is based on the stochastic differential equation

$$dx_t = 10x_t(1 - x_t^2)dt + dw_t, \quad 0 \le t \le 1, \quad (30)$$

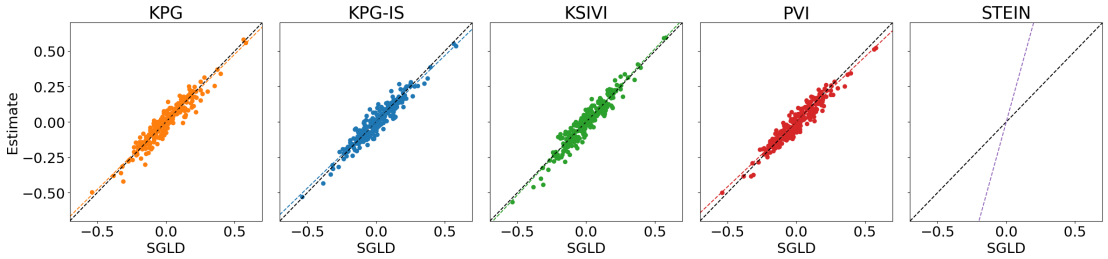

Figure 2: A scatter plot of all pairwise correlation coefficients $\rho_{i,j}$ between our estimates and those obtained from SGLD. The identity line indicates perfect agreement.

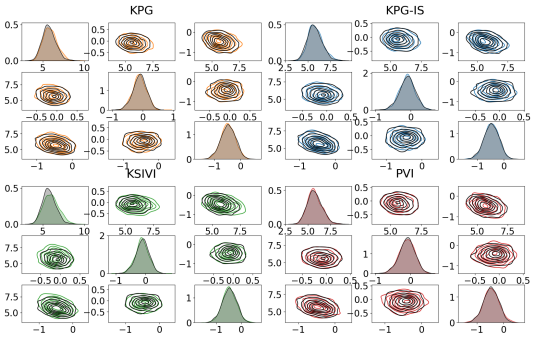

Figure 3: Comparison of marginal and pairwise density estimates for $\boldsymbol{\beta}^{(1)}$, $\boldsymbol{\beta}^{(2)}$, and $\boldsymbol{\beta}^{(3)}$, with SGLD estimates shown in black for reference.

with $x_0 = 0$ and w_t a standard Brownian motion (Detommaso et al., 2018). After discretization of the SDE using the Euler-Maruyama scheme, one obtained a 100-dimensional latent variable $\boldsymbol{x}$ with prior $p(\boldsymbol{x})$ and observations $\boldsymbol{y}$ obtained by perturbing 20 time points of $\boldsymbol{x}$ using Gaussian noise. Accordingly, the likelihood $p(\boldsymbol{y}|\boldsymbol{x})$ is based on a Gaussian distribution assumption with

$$p(y_i|x_i) = (2\pi\sigma^2)^{-0.5} \exp((2\sigma^2)^{-1}(y_i - x_i)^2) \quad (31)$$

with $\sigma^2 = 0.1$. The goal is to approximate the posterior $p(\boldsymbol{x}|\boldsymbol{y})$. As before, the ground truth is generated using SGLD with 100,000 iterations and 1000 independent particles. The step size is again chosen to 0.0001. We again use the settings suggested for comparison methods as discussed in the literature (Cheng et al., 2024). In addition to PVI and KSIVI, we also run amortized SVGD (STEIN) by Feng et al. (2017).

Results Fig. 4 summarizes the results by showing the sample path and estimated confidence interval of each method. Using SGLD as a ground truth, we see that most methods perform well. However, notably less variation in the posterior is observed for PVI and also for STEIN. KPG-IS, KPG, and KSIVI perform similarly well, demonstrating that our method is on par with the current state-of-the-art. Fig. 1 exemplarily depicts one characteristic runtime comparison between KSIVI and KPG-IS. While both methods reach the same value, KPG is notably faster in convergence (cf. Fig. 1 and Table 1). Further replications of this phenomenon can be found in the Appendix D.2.

Table 1: Log marginal likelihood estimates on the conditional diffusion benchmark. The reference estimate is computed using 1000 high-quality SGLD samples, while each method's estimate is based on 60,000 samples.

Method	↑ Log ML	↓ Seconds per iteration
KPG-IS	**74528**	1.45×10^{-2}
KPG	74311	2.50×10^{-3}
KSIVI	74504	1.40×10^{-2}
STEIN	70371	2.50×10^{-3}
PVI	47853	5.00×10^{-3}

5.2 Bayesian Linear Regression

Another commonly used benchmark experiment for SIVI methods in moderate dimensions is a Bayesian logistic regression on the WAVEFORM dataset, which can be obtained from the UCI repository (Dua and Graff, 2017). This problem was initially proposed in Yin and Zhou (2018). Given $y_i \in \{0, 1\}, i = 1, \ldots, N$ with $N = 400$ and features $\boldsymbol{x}_i \in \mathbb{R}^{21}$, the model is defined by likelihood

$$p(\boldsymbol{y}|\boldsymbol{X}; \boldsymbol{\beta}) = \prod_{i=1}^{N} \exp(\boldsymbol{x}_i^\top \boldsymbol{\beta})^{y_i} (1 - \exp(\boldsymbol{x}_i^\top \boldsymbol{\beta}))^{1-y_i},$$

where the goal is inference for the latent variable $\boldsymbol{\beta} \in \mathbb{R}^{22}$. As prior $p(\boldsymbol{\beta})$ an uninformative normal distribution with mean zero and variance 100 is used.

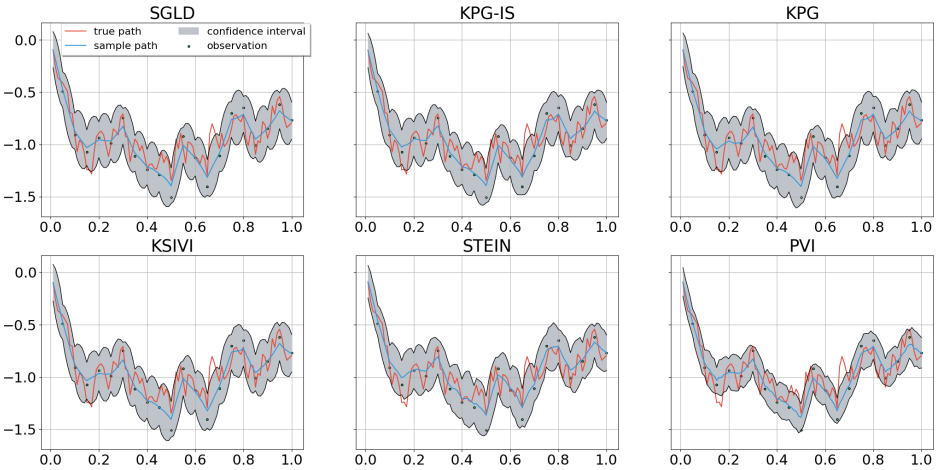

Figure 4: Comparison of posterior quality of different models (facets) for the diffusion process.

The ground truth for this example is obtained by simulating stochastic gradient Langevin dynamics (SGLD) (Welling and Teh, 2011). Following Cheng et al. (2024), we use 400,000 iterations and 1000 samples for SGLD with a step size of 0.0001. We use the setup and optimized hyperparameters as in Cheng et al. (2024) to ensure a fair comparison.

Results

Figures 2 and 3 show that all SIVI methods perform similarly in this example, with the exception of amortized SVGD, which diverges under the exact same hyperparameters used for KPG. This divergence, shown in the Appendix D.1, highlights the impact of higher variance in the score gradient estimates. No systematic over- or underestimation of variances and correlations can be observed among the remaining methods. For KPG-IS, we set $\underline{\alpha} = 0.99$ and reused the uninformed $\boldsymbol{\epsilon}$-samples, as detailed in Appendix E, to maintain computational efficiency, as target density evaluations are significantly more expensive in this example compared to the previous one.

5.3 Bayesian Neural Networks

Setup We add a Bayesian neural network (BNN) regression benchmark to assess sample efficiency and training speed on small- and medium-scale UCI-style datasets. To ensure comparability, we adopt the experimental setup introduced in Cheng et al. (2024), using the same architectures, optimizers, and data splits. All reported results are averaged over three independent runs, with means and standard deviations provided.

Table 2: BNN regression benchmarks. Best per metric and dataset in **bold**.

Dataset	Method	NLL ↓	RMSE ↓	it/s ↑
boston	KPG	$\mathbf{2.47 \pm 0.0113}$	$\mathbf{2.56 \pm 0.0729}$	**220**
	KPGIS	2.49 ± 0.00690	2.67 ± 0.0321	105
	KSIVI	2.48 ± 0.00395	2.57 ± 0.0361	123
concrete	KPG	3.51 ± 0.0260	7.55 ± 0.168	**240**
	KPGIS	3.48 ± 0.0537	7.52 ± 0.140	107
	KSIVI	$\mathbf{3.42 \pm 0.0598}$	$\mathbf{7.30 \pm 0.285}$	130
yacht	KPG	$\mathbf{0.808 \pm 0.00410}$	$\mathbf{0.205 \pm 0.00744}$	**360**
	KPGIS	0.815 ± 0.0190	0.212 ± 0.0333	128
	KSIVI	0.859 ± 0.0464	0.233 ± 0.0233	160

Results Across datasets, KPG attains strong predictive performance while training notably faster (higher iterations per second). KSIVI matches or slightly surpasses the best likelihood/RMSE on *concrete*, but trains slower than KPG. These results align with our main claims about computational efficiency. KPG-IS remains competitive but does not outperform KPG or KSIVI here; in this higher-dimensional setting, where comparisons rely on predictive metrics rather than known ground truth, its advantage appears attenuated.

6 DISCUSSION

Our results demonstrate that semi-implicit variational inference with kernelized path gradients consistently outperforms amortized Stein variational methods across benchmarks, highlighting the b enefits of explicitly leveraging the semi-implicit structure. The variance reduction achieved by KPG translates into

more stable optimization and more accurate posterior approximations, particularly in challenging high-dimensional settings. Compared to the previous state-of-the-art method, KSIVI, our approach achieves comparable performance with significantly improved computational efficiency, making it a practical and scalable choice for complex models.

Limitations

The proposed method does not inherently promote exploration, which may limit its effectiveness in capturing complex distributions. While this is a common limitation shared with many related approaches, it remains an important area for improvement. In principle, our method can be combined with techniques such as temperature annealing (Rezende and Mohamed, 2015) to encourage better mode coverage, but a more principled and integrated exploration mechanism would be desirable. Addressing this limitation is a promising direction for future research.

References

Ziheng Cheng, Longlin Yu, Tianyu Xie, Shiyue Zhang, and Cheng Zhang. Kernel semi-implicit variational inference. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=w5oUo0LhO1.

Gianluca Detommaso, Tiangang Cui, Alessio Spantini, Youssef Marzouk, and Robert Scheichl. A stein variational newton method. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9187–9197, Red Hook, NY, USA, 2018. Curran Associates Inc.

Dheeru Dua and Casey Graff. Uci machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Yihao Feng, Dilin Wang, and Qiang Liu. Learning to draw samples with amortized stein variational gradient descent. In Gal Elidan, Kristian Kersting, and Alexander Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017. URL http://auai.org/uai2017/proceedings/papers/206.pdf.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Yingzhen Li and Richard E. Turner. Gradient estimators for implicit models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJi9WOeRb.

Jen Ning Lim and Adam Michael Johansen. Particle semi-implicit variational inference. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=p3gMGkHMkM.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/b3ba8f1bee1238a2f37603d90b58898d-Paper.pdf.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/rezende15.html.

Geoffrey Roeder, Yuhuai Wu, and David K Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/e91068fff3d7fa1594dfdf3b4308433a-Paper.pdf.

Artem Sobolev and Dmitry P Vetrov. Importance weighted hierarchical variational inference. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5737c6ec2e0716f3d8a7a5c4e0de0d9a-Paper.pdf.

Michalis K. Titsias and Francisco Ruiz. Unbiased implicit variational inference. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 167–176. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/titsias19a.html.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *International Conference on Machine Learning*, 2011. URL https://api.semanticscholar.org/CorpusID:2178983.

Mingzhang Yin and Mingyuan Zhou. Semi-implicit variational inference. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5660–5669. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/yin18b.html.

Longlin Yu and Cheng Zhang. Semi-implicit variational inference via score matching. In *The Eleventh Inter-*

national Conference on Learning Representations, 2023.
URL https://openreview.net/forum?id=sd90a2ytrt.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes – see Sections 3, 2]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes – variance and complexity analyses in Section 3]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes – provided in the supplementary material with dependencies listed]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes – see Propositions 3.1–3.3]
   (b) Complete proofs of all theoretical results. [Yes – full proofs in Appendix A]
   (c) Clear explanations of any assumptions. [Yes – explained alongside the propositions in Section 3]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes – code provided in the supplementary material]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes – described in Appendix D]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes – definitions and repetitions are stated in Section 5 and Appendix B]
   (d) A description of the computing infrastructure used (e.g., type of GPUs, internal cluster, or cloud provider). [Yes – see Section 5 and Appendix C]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes – datasets and baselines cited, e.g., Cheng et al. (2024), Dua and Graff (2017)]
   (b) The license information of the assets, if applicable. [Yes – all software used is open source]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable – no new assets created]
   (d) Information about consent from data providers/curators. [Not Applicable – only public datasets used]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable – no such data involved]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable – no human subjects involved]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable – no human subjects involved]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable – no human subjects involved]

Supplementary Materials

A PROOFS

A.1 Variance Comparison of Score Gradient Estimators

First note since $\mathbb{E}\left[\hat{s}_{\text{STEIN},k}(\boldsymbol{z})\right] = \mathbb{E}\left[\hat{s}_{\text{SI},k}(\boldsymbol{z})\right]$ it follows that

$$\begin{aligned}
\Delta\mathbb{V} &:= \operatorname{trace}\left(\mathbb{V}\left[\hat{s}_{\text{STEIN},k}(\boldsymbol{z})\right] - \mathbb{V}\left[\hat{s}_{\text{SI},k}(\boldsymbol{z})\right]\right) && (32)\\
&= \operatorname{trace}\left(\mathbb{E}\left[\hat{s}_{\text{STEIN},k}(\boldsymbol{z})\hat{s}_{\text{STEIN},k}(\boldsymbol{z})^\top\right] - \mathbb{E}\left[\hat{s}_{\text{SI},k}(\boldsymbol{z})\hat{s}_{\text{SI},k}(\boldsymbol{z})^\top\right]\right) && (33)\\
&= \mathbb{E}\left[\hat{s}_{\text{STEIN},k}(\boldsymbol{z})^\top \hat{s}_{\text{STEIN},k}(\boldsymbol{z})\right] - \mathbb{E}\left[\hat{s}_{\text{SI},k}(\boldsymbol{z})^\top \hat{s}_{\text{SI},k}(\boldsymbol{z})\right] && (34)\\
&= \mathbb{E}\left\|\hat{s}_{\text{STEIN},k}(\boldsymbol{z})\right\|_2^2 - \mathbb{E}\left\|\hat{s}_{\text{SI},k}(\boldsymbol{z})\right\|_2^2. && (35)
\end{aligned}$$

Since for the Gaussian density kernel k

$$\nabla_{\boldsymbol{z}'} k(\boldsymbol{z}, \boldsymbol{z}') = k(\boldsymbol{z}, \boldsymbol{z}') \cdot \frac{\boldsymbol{z}' - \boldsymbol{z}}{\sigma_k^2}, \tag{36}$$

and for the Gaussian conditional likelihood $q_{\boldsymbol{z}|\boldsymbol{\epsilon}}$

$$\nabla_{\boldsymbol{z}'} \log q_{\boldsymbol{z}|\boldsymbol{\epsilon}}(\boldsymbol{z}'|\boldsymbol{\epsilon}') = \operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^{-2}\left(\boldsymbol{z}' - \boldsymbol{\mu}_{\boldsymbol{\epsilon}'}\right), \tag{37}$$

it holds that

$$\begin{aligned}
\Delta\mathbb{V} = &\frac{1}{n}\mathbb{E}_{\boldsymbol{z}'\sim q_{\boldsymbol{z}}}\left[k(\boldsymbol{z}, \boldsymbol{z}')^2 \cdot \frac{\|\boldsymbol{z}' - \boldsymbol{z}\|^2}{\sigma_k^4}\right] \\
&- \frac{1}{n}\mathbb{E}_{\boldsymbol{\epsilon}', \boldsymbol{\eta}'\sim p_{\boldsymbol{\epsilon},\boldsymbol{\eta}}}\left[k(\boldsymbol{z}, \operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}' + \boldsymbol{\mu}_{\boldsymbol{\epsilon}'})^2 \cdot \|\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^{-1}\boldsymbol{\eta}'\|_2^2\right]. && (38)\\
= &\frac{1}{n}\mathbb{E}_{\boldsymbol{\epsilon}', \boldsymbol{\eta}'\sim p_{\boldsymbol{\epsilon},\boldsymbol{\eta}}} k(\boldsymbol{z}, \operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}' + \boldsymbol{\mu}_{\boldsymbol{\epsilon}'})^2 \\
&\cdot \left[\frac{\|\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}' + \boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2}{\sigma_k^4} - \|\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^{-1}\boldsymbol{\eta}'\|_2^2\right]. && (39)
\end{aligned}$$

A.2 Sufficient Condition for Lower Score Gradient Variance

Since

$$\Delta\mathbb{V} = \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}', \boldsymbol{\eta}'\sim p_{\boldsymbol{\epsilon},\boldsymbol{\eta}}} \underbrace{\frac{k(\boldsymbol{z}, \operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}' + \boldsymbol{\mu}_{\boldsymbol{\epsilon}'})^2}{n}}_{\geq 0} \cdot \underbrace{\left[\frac{\|\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}' + \boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2}{\sigma_k^4} - \|\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^{-1}\boldsymbol{\eta}'\|_2^2\right]}_{=:\beta}, \tag{40}$$

it follows from $\beta \geq 0$ a.s. that $\Delta\mathbb{V} \geq 0$.

From this, we get that

$$\begin{aligned}
\beta \geq 0 \text{ a.s.} &\iff \frac{\|\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}' + \boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2}{\|\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^{-1}\boldsymbol{\eta}'\|_2^2} \geq \sigma_k^4 \text{ a.s.} && (41)\\
&\Leftarrow \frac{\|\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}'\|_2^2 + 2(\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}')^\top(\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}) + \|\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2}{\frac{1}{\min(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^2}\|\boldsymbol{\eta}'\|_2^2} \geq \sigma_k^4 \text{ a.s.} && (42)\\
&\Leftarrow \frac{\min(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^2\|\boldsymbol{\eta}'\|_2^2 - 2\|(\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}')\|_2\|(\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z})\|_2 + \|\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2}{\frac{1}{\min(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^2}\|\boldsymbol{\eta}'\|_2^2} \geq \sigma_k^4 \text{ a.s.} && (43)\\
&\Leftarrow \min(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^4 - 2\min(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^2 \max(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\frac{\|\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2}{\|\boldsymbol{\eta}'\|_2} + \min(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^2\frac{\|\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2}{\|\boldsymbol{\eta}'\|_2^2} \geq \sigma_k^4 \quad \text{a.s.} && (44)
\end{aligned}$$

A.3 Upper Bound for the Difference in Score Gradient Variance

Under the assumptions that $\alpha = k(\boldsymbol{z}, \boldsymbol{z}')^2$ and $\beta = \left[\frac{\|\mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}' + \boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2}{\sigma_k^4} - \|\mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^{-1}\boldsymbol{\eta}'\|_2^2\right]$ are negatively correlated, it holds that

$$
\begin{aligned}
\Delta\mathbb{V} &= \mathbb{E}\left[\alpha\beta\right] && (45)\\
&\leq \mathbb{E}\left[\alpha\right] \cdot \mathbb{E}\left[\beta\right] && (46)\\
&= \frac{1}{n}\mathbb{E}_{\boldsymbol{\epsilon}', \boldsymbol{\eta}' \sim p_{\boldsymbol{\epsilon},\boldsymbol{\eta}}} k(\boldsymbol{z}, \mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}' + \boldsymbol{\mu}_{\boldsymbol{\epsilon}'})^2 \\
&\quad \cdot \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}', \boldsymbol{\eta}' \sim p_{\boldsymbol{\epsilon},\boldsymbol{\eta}}} \left[\frac{\|\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'} \odot \boldsymbol{\eta}'\|_2^2 + 2(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'} \odot \boldsymbol{\eta}')^\top(\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}) + \|\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2}{\sigma_k^4} - \|\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'}^{\odot -1} \odot \boldsymbol{\eta}'\|_2^2\right] && (47)\\
&\overset{\boldsymbol{\eta}\sim\mathcal{N}(0,I)}{=} \frac{1}{n}\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{z}}} k(\boldsymbol{z}, \boldsymbol{z}')^2 \\
&\quad \cdot \underbrace{\left(\frac{\mathbb{E}_{\boldsymbol{\epsilon}'\sim p_{\boldsymbol{\epsilon}}}\left[\|\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'}\|_2^2 + \|\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2\right]}{\sigma_k^4} - \mathbb{E}_{\boldsymbol{\epsilon}'\sim p_{\boldsymbol{\epsilon}}}\left[\|\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'}^{\odot -1}\|_2^2\right]\right)}_{=:\gamma} \cdot && (48)\\
&= \frac{1}{n}\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{z}}}\left[\frac{1}{(2\pi\sigma_k^2)^{d_z}} \exp\left(-\frac{\|\boldsymbol{z} - \boldsymbol{z}'\|_2^2}{\sigma_k^2}\right)\right] \cdot \gamma && (49)\\
&\approx \frac{q_{\boldsymbol{z}}(\boldsymbol{z})}{n(2\sqrt{\pi}\sigma_k)^{d_z}} \cdot \gamma. && (50)
\end{aligned}
$$

A.4 Bounds of the Optimal Proposal Distribution

For $\alpha(\boldsymbol{z}) = 1$, we get the trivial upper bound solution

$$\tau^*_{\boldsymbol{\epsilon}|\boldsymbol{z}} = 1 \cdot p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}) + 0 \cdot \widetilde{\tau}^*_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z}) = p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}). \tag{51}$$

For $\alpha(\boldsymbol{z}) = 0$, assuming that $\widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}}$ is sufficiently flexible, we get that

$$
\begin{aligned}
&\widetilde{\tau}^*_{\boldsymbol{\epsilon}|\boldsymbol{z}} \in \arg\min_{\widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}}} \mathbb{E}_{\boldsymbol{z},\boldsymbol{\epsilon}\sim q_{\boldsymbol{z},\boldsymbol{\epsilon}}}\left[-\log\left(\tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z})q_{\boldsymbol{z}}(\boldsymbol{z})\right)\right] && (52)\\
\iff &\widetilde{\tau}^*_{\boldsymbol{\epsilon}|\boldsymbol{z}} \in \arg\min_{\widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}}} \mathbb{E}_{\boldsymbol{z},\boldsymbol{\epsilon}\sim q_{\boldsymbol{z},\boldsymbol{\epsilon}}}\left[-\log\left(\widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}|\boldsymbol{z})q_{\boldsymbol{z}}(\boldsymbol{z})\right)\right] && (53)\\
\iff &\widetilde{\tau}^*_{\boldsymbol{\epsilon}|\boldsymbol{z}} \in \arg\min_{\widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}}} D_{\mathrm{KL}}(q_{\boldsymbol{z},\boldsymbol{\epsilon}} \| \widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}} \cdot q_{\boldsymbol{z}}) && (54)\\
\iff &\widetilde{\tau}^*_{\boldsymbol{\epsilon}|\boldsymbol{z}} \cdot q_{\boldsymbol{z}} = q_{\boldsymbol{z},\boldsymbol{\epsilon}} && (55)\\
\iff &\widetilde{\tau}_{\boldsymbol{\epsilon}|\boldsymbol{z}} = q_{\boldsymbol{\epsilon}|\boldsymbol{z}} \quad (= \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}). && (56)
\end{aligned}
$$

B FURTHER BENCHMARKS

We evaluate all kernelized SIVI variants on the *banana*, *x-shaped*, and *multimodal* benchmark distributions, originally proposed by Cheng et al. (2024), with the corresponding target densities summarized in Table 3. We train each model for 500 epochs with 100 optimization steps per epoch and a batch size of 500. The architecture is a fully connected ReLU network with a latent dimension of 3, hidden layer size of 50, and output dimension of 2. Optimization is performed using the Adam optimizer with a learning rate of 0.001. A learning rate decay with factor 0.9 is applied every 1000 steps. For the *x-shaped* and *banana* targets, no annealing is used, while for the *multimodal* target, annealing is enabled.

The performance across three independent runs is presented in Table 4, and representative contour plots of the final models are shown in Figure 5. Among the tested methods, KPG-IS consistently outperforms all other kernel-based SIVI approaches. This is particularly evident on the banana benchmark, where the comparison with KSIVI highlights KPG-IS's ability to more effectively capture narrow, high-density regions. Additionally, the comparison between Stein and KPG-IS clearly demonstrates the benefits of variance reduction through structured exploitation of the SIVI framework.

Table 3: Definitions of the benchmark target distributions.

Benchmark	Distribution	Parameters
Banana	$z = (\nu_1, \nu_1^2 + \nu_2 + 1)^\top$, where $\nu \sim \mathcal{N}(0, \Sigma)$	$\Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}$
Multimodal	$z \sim \frac{1}{2}\mathcal{N}(\mu_1, I) + \frac{1}{2}\mathcal{N}(\mu_2, I)$	$\mu_1 = (-2, 0)^\top$, $\mu_2 = (2, 0)^\top$
X-shaped	$z \sim \frac{1}{2}\mathcal{N}(0, \Sigma_1) + \frac{1}{2}\mathcal{N}(0, \Sigma_2)$	$\Sigma_1 = \begin{bmatrix} 2 & 1.8 \\ 1.8 & 2 \end{bmatrix}$, $\Sigma_2 = \begin{bmatrix} 2 & -1.8 \\ -1.8 & 2 \end{bmatrix}$

Table 4: Benchmark comparison for problems in Table 3. Each negative log likelihood (NLL) is computed on 100,000 samples from the data-generating process evaluated using model estimates from 100,000 ϵ-samples. Reported are mean ± standard deviation over 3 runs. Best-performing methods (closest to DGP) are shown in **bold**.

Dataset	DGP (Ground Truth)	KPG-IS	KPG	KSIVI	STEIN
banana	2.0024	$\mathbf{2.0913} \pm 0.0026$	2.1295 ± 0.0341	3.6227 ± 0.2545	8.1287 ± 2.4536
x_shaped	3.1219	$\mathbf{3.5973} \pm 0.8169$	4.0132 ± 0.7670	3.8713 ± 1.2980	7.9521 ± 0.0980
multimodal	3.4663	$\mathbf{3.4666} \pm 0.0001$	3.4703 ± 0.0010	3.4671 ± 0.0001	7.8133 ± 0.5641

C COMPUTATIONAL ENVIRONMENT

All experiments are performed on a Linux-based server A5000 server with 2 GPUs, 24GB VRAM, and Intel Xeon Gold 5315Y processor with 3.20 GHz.

D EXPERIMENTAL DETAILS

For all kernelized SIVI variants, we use the Gaussian kernel and a conditional Gaussian likelihood with diagonal covariance. The mean of the likelihood is given by the output of a fully connected neural network, which we refer to as the SIVI model, while the variance is represented by a learnable parameter vector. We build upon the public KSIVI (Cheng et al., 2024) implementation (`https://github.com/longinyu/ksivi`) and reuse parts of their evaluation framework to ensure fair and directly comparable results.

D.1 Bayesian Logistic Regression

We used a batch size of 100 for training, and the likelihood was evaluated using the full dataset without subsampling. The model is a fully connected ReLU network with a latent dimension of 10, hidden layer size of 100, and output dimension of 22. Both the parameters of the SIVI models and their variance vector of the conditional Gaussian likelihood were optimized using the Adam optimizer with a learning rate of 0.001. A learning rate decay with factor 0.9 was applied every 3000 steps. Training was performed for 200,000 iterations.

We present representative marginal and pairwise distributions of the first three components for the Stein variant in Figure 6, demonstrating that it fails to converge to the correct solution.

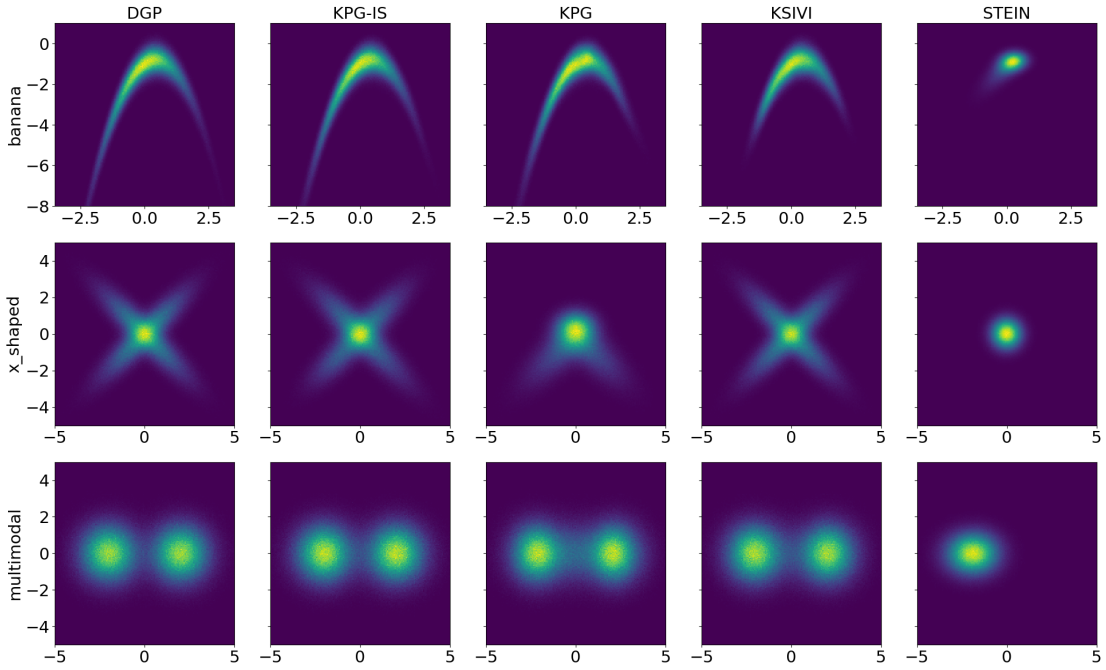

Figure 5: Density contour plots for the DGP and each method from one representative repetition. Each plot visualizes the estimated density using 100,000 $\boldsymbol{\epsilon}$-samples. The DGP serves as the reference distribution, while the others illustrate the approximation quality of each method.

D.2 Conditional Diffusion Process

We trained the model for 1000 epochs with 100 optimization steps per epoch, using a batch size of 128. The model is a fully connected ReLU network with a latent dimension of 100, hidden layer size of 128, and output dimension of 100. Both the parameters of the SIVI models and their variance vector of the conditional Gaussian likelihood were optimized using the Adam optimizer with a learning rate of 0.0002. A learning rate decay with factor 0.9 was applied every 10,000 steps.

We repeat the 100-dimensional conditional diffusion process benchmark three times for the two best-performing models, KSIVI and KPG-IS, highlighting their stable training dynamics. As shown in Figure 7, KPG-IS consistently demonstrates greater computational efficiency compared to KSIVI.

E Implementation Details

Note that each iteration of KPG requires l target density evaluations, KSIVI requires $2l$, and KPG-IS requires l^2. When the cost of evaluating the target density becomes substantial, the quadratic cost of KPG-IS may become prohibitive. To address this, we employ the following adapted version of the KPG-IS algorithm:

In each iteration, we sample $\boldsymbol{\epsilon}_j \sim p_{\boldsymbol{\epsilon}}$ for $j = 1, \dots, l$, and reuse these samples in place of $\boldsymbol{\epsilon}_{i,j} \sim p_{\boldsymbol{\epsilon}}$ wherever they would appear in Algorithm 2. By setting $\underline{\alpha}$ close to one, we achieve a substantial computational efficiency gain while still benefiting from the performance improvements afforded by partially informed sampling.

For further implementation details, we refer the reader to our code repository at `https://github.com/tpielok/KPG`.

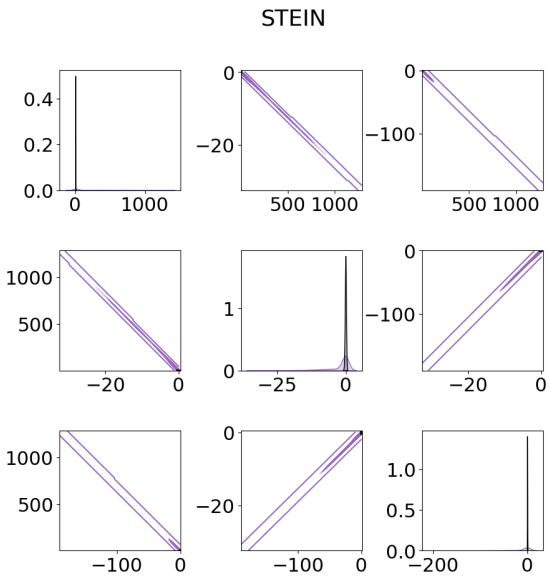

Figure 6: STEIN: Marginal and pairwise density estimates for $\boldsymbol{\beta}^{(1)}$, $\boldsymbol{\beta}^{(2)}$, and $\boldsymbol{\beta}^{(3)}$, with SGLD estimates shown in black for reference.

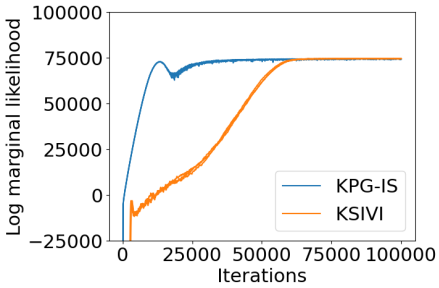

Figure 7: Convergence speed comparison between the current state-of-the-art method KSIVI and our proposal KPG-IS based on 3 repetitions.

Part IV.

Conclusion

6. Concluding Remarks

This thesis has addressed fundamental challenges in variational inference through three distinct but interconnected contributions, each advancing the state of the art in approximating complex probability distributions. Specifically, this work explores two major modern directions in variational inference: functional variational inference, which operates directly in function space, and semi-implicit variational inference, which combines the tractability of explicit methods with the expressiveness of implicit distributions.

Common Themes and Insights. Several unifying themes emerge across these contributions.

First, two of the works use *kernel methods* and RKHS theory to achieve computational efficiency. Whether solving for optimal functional perturbations in function space, avoiding explicit score estimation, or exploiting the kernel expectation trick for semi-implicit models, RKHS-based formulations provide both theoretical guarantees and practical tractability.

Second, although in the broader machine learning literature separate optimization of *auxiliary models* is often considered problematic due to instability (Li and Turner, 2018), our situation differs in important ways. Here, we benefit from stable gradients when optimizing the proposal distribution, and our estimator remains consistent for any fixed proposal. This means that, contrary to common concerns about separate optimization, in our context it can be more scalable and practical than joint optimization and does not inherit the typical disadvantages found in other multi-model training setups.

Third, an important insight is that *path gradient estimators* (Roeder et al., 2017) allow (semi-)implicit variational inference to reduce to estimating the score (gradient of the log-density) of the variational model. This subproblem is typically much easier than matching the full variational distribution to the target, particularly when the variational density is intractable or unavailable. Additionally, we note that for SIVI, the path gradient estimator is more computationally efficient than the standard reparameterization gradient estimator, since it avoids expensive backpropagation through the score gradient, which is given by an inner expectation.

Impact on Variational Inference. The contributions of this thesis advance variational inference in several important ways:

- **Improved Computational Efficiency:** In Chapter 5, we approximately halved the sample complexity compared to the state-of-the-art KSIVI method by introducing the kernelized path gradient (KPG) estimator. Alongside this, the elimination of inner MCMC loops in Chapter 4 and the avoidance of explicit score estimation in Chapter 3 both contribute to making advanced VI methods more practical for large-scale applications.

- **Better Theoretical Understanding:** The connections between amortized SVGD, kernel methods, and semi-implicit VI clarified in Chapter 5, along with the consistency guarantees for the importance sampling framework in Chapter 4, deepen our understanding of gradient estimation in hierarchical variational models.

- **Methodological Flexibility:** The successful integration of conditional normalizing flows into the SIVI framework demonstrates that modern variational inference approaches can be effectively combined with classical generative models.

Limitations and Trade-offs. The contributions of this thesis represent significant advances, but they also come with limitations and trade-offs:

While we include Bayesian neural network experiments as part of our evaluation, their evidential value for assessing posterior approximation quality is inherently limited. In such settings, the true posterior is unavailable, making it impossible to directly quantify how well a variational method matches it. Nevertheless, these experiments are often expected in the literature, and for functional VI approaches they remain one of the few practical ways to obtain at least indirect empirical insights. Still, their prominence is arguably overstated: evaluating whether a method yields good predictive performance is related, but largely orthogonal to evaluating the match of the inferred posterior to the true posterior. This motivates complementary evaluations in high-dimensional synthetic but challenging settings where ground-truth distributions are known, enabling rigorous and unambiguous assessment of inference quality.

Also, all three contributions employ one-shot generators, which are computationally very efficient for sampling, as they generate samples in a single forward pass. However, recent literature has demonstrated that more costly iterative procedures, such as diffusion-based or MCMC-augmented methods, can produce higher-quality samples at the expense of increased computational cost (Vargas et al., 2024; Akhound-Sadegh et al., 2024). The trade-off between sample quality and computational efficiency remains an important consideration when selecting inference methods for specific applications.

A further consideration is that all three contributions are based on minimizing the reverse Kullback-Leibler divergence. This choice brings important theoretical guarantees: minimizing the reverse KL implies convergence in total variation when the variational family is sufficiently expressive. However, as highlighted in Section 2.4, the reverse KL is also mode-seeking in practice. This means that the approximation usually has no strong incentive to explore the regions outside its current support. While this behavior is not a fundamental flaw, it is a practical caveat to be aware of, especially if the target distribution has weakly connected probability mass and full exploration is important for a given application. In such settings, additional strategies may be required to encourage broader exploration, as discussed in the introduction.

More generally, all three contributions inherit the general limitations of variational inference: the quality of approximation is constrained by the chosen variational family, local optima can trap optimization, and there are no universal guarantees on the approximation error. Nevertheless, the empirical performance across various benchmarks suggests that these methods represent meaningful progress toward closing the gap between variational inference and MCMC in terms of accuracy, while maintaining computational advantages.

Broader Context. This thesis advances the practical use of Bayesian inference for complex, high-dimensional settings by contributing new methods in variational inference. While variational inference is widely adopted for its scalability and compatibility with modern optimization methods, closing the gap in accuracy compared to MCMC remains a key challenge.

The turn toward semi-implicit and functional variational inference reflects a broader evolution in the field: as the shortcomings of mean-field and traditional flow models have become clear, researchers have sought out more expressive yet tractable frameworks. This work pushes this transition forward by showing how modern tools from RKHS theory, importance sampling, and advanced gradient estimators can address specific limitations of current approaches.

A key open challenge that emerges from this perspective is the construction of low-MSE estimators for the score gradient of (semi-)implicit variational distributions with only linear sampling cost. Achieving this would represent substantial progress toward fully practical VI in high-dimensional settings. Although this thesis does not yet reach that ideal, AISIVI (see Chapter 4) advances the theoretical pathway toward it, and the KPG estimator (see Chapter 5) demonstrates that low-MSE score gradients are attainable, albeit currently requiring a quadratic number of samples. Developing estimators that combine this accuracy with linear complexity remains an important direction for future research.

Contributing Publications

Tobias Pielok, Bernd Bischl, and David Rügamer. 2023. Approximate Bayesian Inference with Stein Functional Variational Gradient Descent. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/pdf?id=a2-aoqmeYM4.

Tobias Pielok, Bernd Bischl, and David Rügamer. 2025. Revisiting Unbiased Implicit Variational Inference. In *International Conference on Machine Learning (ICML)*. https://openreview.net/pdf?id=Fm1K8tMlaf.

Tobias Pielok, Bernd Bischl, and David Rügamer. 2026. Semi-Implicit Variational Inference via Kernelized Path Gradient Descent. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*. https://arxiv.org/abs/2506.05088.

Further References

S. Agapiou, O. Papaspiliopoulos, D. Sanz-Alonso, and A. M. Stuart. 2017. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, 32(3):405 – 431.

Tara Akhound-Sadegh, Jarrid Rector-Brooks, Joey Bose, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcin Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, Nikolay Malkin, and Alexander Tong. 2024. Iterated denoising energy matching for sampling from Boltzmann densities. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 760–786. PMLR.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. 2019. A convergence theory for deep learning via over-parameterization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 242–252. PMLR.

Aloisio Araujo and Evarist Giné. 1980. *The central limit theorem for real and Banach valued random variables*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, New York.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR.

Nachman Aronszajn. 1950. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404.

James O. Berger. 1985. *Statistical decision theory and Bayesian analysis*. Springer series in statistics. Springer, New York.

Alain Berlinet and Christine Thomas-Agnan. 2004. *RKHS and stochastic processes*, pages 55–108. Springer US, Boston, MA.

Michael Betancourt. 2018. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*.

Patrick Billingsley. 1995. *Probability and measure*, 3rd edition. Wiley.

Bernd Bischl, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, Theresa Ullmann, Marc Becker, Anne-Laure Boulesteix, Difan Deng, and Marius Lindauer. 2023. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data Mining and Knowledge Discovery*, 13(2):e1484.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. 2017. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Vladimir I. Bogachev. 1998. *Gaussian measures*, volume 62 of *Mathematical Surveys and Monographs*. American Mathematical Society.

Claudio Carmeli, Ernesto De Vito, and Alessandro Toigo. 2006. Vector valued reproducing kernel Hilbert spaces of integrable functions and Mercer theorem. *Analysis and Applications*, 4(4):377–408.

Kacper Chwialkowski, Heiko Strathmann, and Arthur Gretton. 2016. A kernel test of goodness of fit. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2606–2615, New York, New York, USA. PMLR.

Thomas M. Cover and Joy A. Thomas. 2006. *Elements of information theory*, 2nd edition. Wiley.

Imre Csiszár. 1967. Information-type measures of difference of probability distributions and indirect observations. *Studia Scientiarum Mathematicarum Hungarica*, 2:299–318.

Klaus Deimling. 1985. *Topological degree in infinite dimensions*, pages 35–94. Springer Berlin Heidelberg, Berlin, Heidelberg.

Joseph Diestel and Jr. Uhl, John J. 1977. *Vector measures*, volume 15 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2017. Density estimation using real NVP. In *International Conference on Learning Representations*.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. 2019. Neural spline flows. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Yihao Feng, Dilin Wang, and Qiang Liu. 2017. Learning to draw samples with amortized Stein variational gradient descent. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*.

Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. 2018. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

G.B. Folland. 2013. *Real analysis: Modern techniques and their applications*. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts. Wiley.

David Freedman. 1999. Wald lecture: On the Bernstein-von Mises theorem with infinite-dimensional parameters. *The Annals of Statistics*, 27(4):1119 – 1141.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. 2013. *Bayesian data analysis*, third edition. edition. Chapman & Hall/CRCtexts in statistical science series. Chapman and Hall/CRC, an imprint of Taylor and Francis, Boca Raton, FL.

Jackson Gorham and Lester Mackey. 2017. Measuring sample quality with kernels. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1292–1301. PMLR.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773.

Jiajun He, Wenlin Chen, Mingtian Zhang, David Barber, and José Miguel Hernández-Lobato. 2025. Training neural samplers with reverse diffusive KL divergence. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 5167–5175. PMLR.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. 2012. Neural networks for machine learning. Lecture 6a, Overview of mini-batch gradient descent.

Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. 2018. Neural autoregressive flows. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2078–2087. PMLR.

Ferenc Huszár. 2017. Variational inference using implicit distributions. *arXiv preprint arXiv:1702.08235*.

Aapo Hyvärinen. 2005. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709.

Pavel Izmailov, Sharad Vikram, Matthew D. Hoffman, and Andrew Gordon Wilson. 2021. What are Bayesian neural network posteriors really like? In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4629–4640. PMLR.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. 1998. *An Introduction to Variational Methods for Graphical Models*, pages 105–161. Springer Netherlands, Dordrecht.

Florian Karl, Tobias Pielok, Julia Moosbauer, Florian Pfisterer, Stefan Coors, Martin Binder, Lennart Schneider, Janek Thomas, Jakob Richter, Michel Lang, Eduardo C. Garrido-Merchán, Juergen Branke, and Bernd Bischl. 2023. Multi-objective hyperparameter optimization in machine learning—an overview. *ACM Trans. Evol. Learn. Optim.*, 3(4).

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Durk P Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Anna Korba, Pierre-Cyril Aubin-Frankowski, Szymon Majewski, and Pierre Ablin. 2021. Kernel Stein discrepancy descent. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5719–5730. PMLR.

L.D. Landau and E.M. Lifshitz. 1980. Chapter i - the fundamental principles of statistical physics. In L.D. Landau and E.M. Lifshitz, editors, *Statistical Physics (Third Edition)*, third edition edition, pages 1–33. Butterworth-Heinemann, Oxford.

Mike Laszkiewicz, Johannes Lederer, and Asja Fischer. 2022. Marginal tail-adaptive normalizing flows. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12020–12048. PMLR.

Michel Ledoux. 2001. *The concentration of measure phenomenon*, volume 89 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI.

Dawei Li, Tian Ding, and Ruoyu Sun. 2022. On the benefit of width for neural networks: Disappearance of basins. *SIAM Journal on Optimization*, 32:1728–1758.

Yingzhen Li and Richard E Turner. 2016. Rényi divergence variational inference. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Yingzhen Li and Richard E. Turner. 2018. Gradient estimators for implicit models. In *International Conference on Learning Representations*.

Jun S. Liu. 2001. *Monte Carlo strategies in scientific computing*. Springer Series in Statistics. Springer, New York.

Qiang Liu, Jason Lee, and Michael Jordan. 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.

Qiang Liu and Dilin Wang. 2016. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

David J. C. MacKay. 1992. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472.

David J C MacKay. 2003. *Information theory, inference and learning algorithms*. Cambridge University Press, Cambridge, England.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. 2017. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2391–2400. PMLR.

Charles A. Micchelli and Massimiliano Pontil. 2005. On learning vector-valued functions. *Neural Computation*, 17(1):177–204.

Laurence Illing Midgley, Vincent Stimper, Gregor N. C. Simm, Bernhard Schölkopf, and José Miguel Hernández-Lobato. 2023. Flow annealed importance sampling bootstrap. In *The Eleventh International Conference on Learning Representations*.

Thomas Minka. 2005. Divergence measures and message passing. Technical report.

Eliakim H. Moore. 1935. General analysis, part i. *Memoirs of the American Philosophical Society*, 1.

Kevin P. Murphy. 2012. *Machine learning: A probabilistic perspective*. The MIT Press.

Radford M. Neal. 1996. *Bayesian learning for neural networks*. Springer New York, New York, NY.

Radford M. Neal. 2011. MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*, pages 113–162. Chapman and Hall/CRC.

Anthony O'Hagan and Jonathan J. Forster. 2004. *Bayesian inference*, 2 edition, volume 2B of *Kendall's Advanced Theory of Statistics*. Arnold, London.

Art B. Owen. 2013. *Monte Carlo theory, methods and examples*. `https://artowen.su.domains/mc/`.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. 2021. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64.

R.K. Pathria and Paul D. Beale. 2022. 1 - the statistical basis of thermodynamics. In R.K. Pathria and Paul D. Beale, editors, *Statistical Mechanics (Fourth Edition)*, fourth edition edition, pages 1–24. Academic Press.

Vern I. Paulsen and Mrinal Raghupathi. 2016. *General theory*, Cambridge Studies in Advanced Mathematics, pages 3–16. Cambridge University Press.

M.S. Pinsker. 1964. *Information and information stability of random variables and processes*. Holden-Day Series in Time Series Analysis. Holden-Day.

Rajesh Ranganath, Sean Gerrish, and David Blei. 2014. Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 814–822, Reykjavik, Iceland. PMLR.

Danilo Rezende and Shakir Mohamed. 2015. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France. PMLR.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 1278–1286, Bejing, China. PMLR.

Christian P. Robert and George Casella. 2004. *Monte Carlo statistical methods*. Springer New York, New York, NY.

Geoffrey Roeder, Yuhuai Wu, and David K Duvenaud. 2017. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Mark J. Schervish. 1995. *Theory of statistics*. Springer New York, New York, NY.

Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. 2001. A generalized representer theorem. In *Computational Learning Theory*, pages 416–426, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jiaxin Shi, Shengyang Sun, and Jun Zhu. 2018. Kernel implicit variational inference. In *International Conference on Learning Representations*.

Ingo Steinwart and Andreas Christmann. 2008. *Kernels and reproducing kernel Hilbert spaces*, pages 110–163. Springer New York, New York, NY.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. 2012. Density-ratio matching under the Bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044.

Dustin Tran, Rajesh Ranganath, and David Blei. 2017. Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Francisco Vargas, Shreyas Padhy, Denis Blessing, and Nikolas Nüsken. 2024. Transport meets variational inference: Controlled Monte Carlo diffusions. In *The Twelfth International Conference on Learning Representations*.

Santosh Vempala and Andre Wibisono. 2019. Rapid convergence of the unadjusted Langevin algorithm: Isoperimetry suffices. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Cédric Villani. 2009. *The Wasserstein distances*, pages 93–111. Springer Berlin Heidelberg, Berlin, Heidelberg.

Christian Walck. 1996. Hand-book on statistical distributions for experimentalists. Technical report, Particle Physics Group, Fysikum, University of Stockholm, Stockholm, Sweden. 1st revision 31 October 1998; last modification 10 September 2007.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256.

Mingzhang Yin and Mingyuan Zhou. 2018. Semi-implicit variational inference. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5660–5669. PMLR.

Kôsaku Yosida. 1995. *Strong convergence and weak convergence*, pages 119–145. Springer Berlin Heidelberg, Berlin, Heidelberg.

Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. 2019. Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):2008–2026.

Shuxin Zheng, Jiyan He, Chang Liu, Yu Shi, Ziheng Lu, Weitao Feng, Fusong Ju, Jiaxi Wang, Jianwei Zhu, Yaosen Min, He Zhang, Shidi Tang, Hongxia Hao, Peiran Jin, Chi Chen, Frank Noé, Haiguang Liu, and Tie-Yan Liu. 2024. Predicting equilibrium distributions for molecular systems with deep learning. *Nature Machine Intelligence*, 6(5):558–567.

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12. Juli 2011, § 8 Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

Während des Überarbeitungsprozesses habe ich Grammarly verwendet, um Rechtschreibfehler und grammatikalische Fehler zu erkennen und zu korrigieren. Außerdem habe ich ChatGPT (GPT-4o) als Hilfsmittel verwendet, um bestimmte einzelne Sätze zu verbessern. Insbesondere habe ich mir Vorschläge zur Aufteilung längerer Sätze und zur Verbesserung der Lesbarkeit generieren lassen. Diese Vorschläge wurden nicht wortwörtlich übernommen, sondern manuell überprüft, angepasst und umformuliert, um sie mit meinem eigenen Schreibstil in Einklang zu bringen.

München, den 26.03.2026 Tobias Pielok