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).

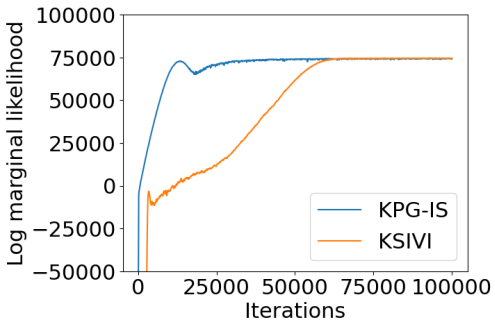

Figure 1: Convergence speed comparison between the current state-of-the-art method KSIVI and our proposal KPG-IS.

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

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

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

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

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.,

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

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

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

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

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

and the semi-implicit gradient estimator

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

we consider the trace of the difference of the covariance matrices

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

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

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

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*

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

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 $\mathbb{E}\left\|\hat{s}_{\mathrm{STEIN},k}(\boldsymbol{z})\right\|_2^2 \geq \mathbb{E}\left\|\hat{s}_{\mathrm{SI},k}(\boldsymbol{z})\right\|_2^2$, 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 $\left[\frac{\|\mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}'+\boldsymbol{\mu}_{\boldsymbol{\epsilon}'}-\boldsymbol{z}\|_2^2}{\sigma_k^4} - \|\mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^{-1}\boldsymbol{\eta}'\|_2^2\right]$ are negatively correlated, it holds that*

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

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 $\left[\frac{\|\mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}'+\boldsymbol{\mu}_{\boldsymbol{\epsilon}'}-\boldsymbol{z}\|_2^2}{\sigma_k^4} - \|\mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^{-1}\boldsymbol{\eta}'\|_2^2\right]$ are negatively correlated 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,
$i = 1, \ldots, m, \quad j = 1, \ldots, l$
repeat
$\boldsymbol{\epsilon}_i \sim p_{\boldsymbol{\epsilon}}, \boldsymbol{\eta}_i \sim p_{\boldsymbol{\eta}}$
$\boldsymbol{z}_i = h_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}_i, \boldsymbol{\eta}_i)$
$\text{loss}_{\text{proposal}} = -1/m \sum_{i=1}^{m} \log \tau_{\boldsymbol{\epsilon}|\boldsymbol{z}}(\boldsymbol{\epsilon}_i | \boldsymbol{z}_i, \alpha(\boldsymbol{z}_i))$
$\boldsymbol{\theta} = \texttt{opt}(\text{loss}_{\text{proposal}}, \boldsymbol{\theta})$

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

$$dx_t = 10x_t(1 - x_t^2)dt + dw_t, \quad 0 \le t \le 1, \quad (30)$$

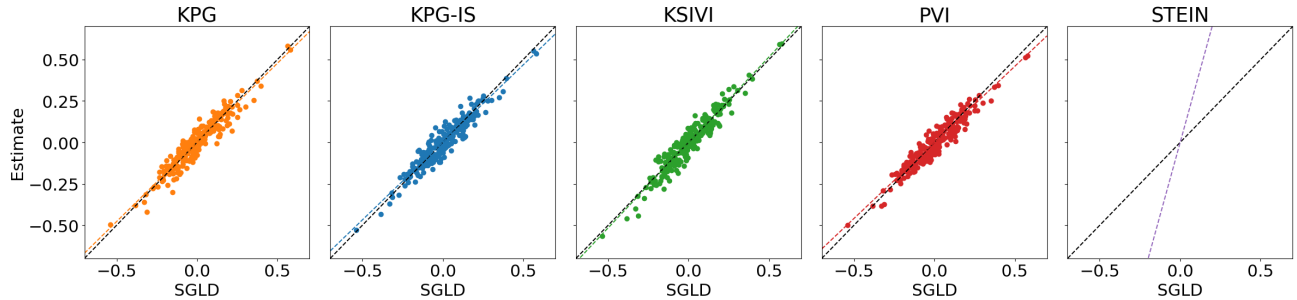

Figure 2: A scatter plot of all pairwise correlation coefficients $\rho_{i,j}$ between our estimates and those obtained from SGLD. The identity line indicates perfect agreement.

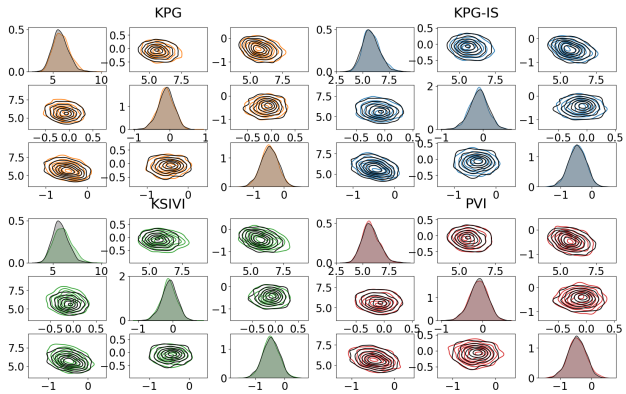

Figure 3: Comparison of marginal and pairwise density estimates for $\boldsymbol{\beta}^{(1)}$, $\boldsymbol{\beta}^{(2)}$, and $\boldsymbol{\beta}^{(3)}$, with SGLD estimates shown in black for reference.

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.

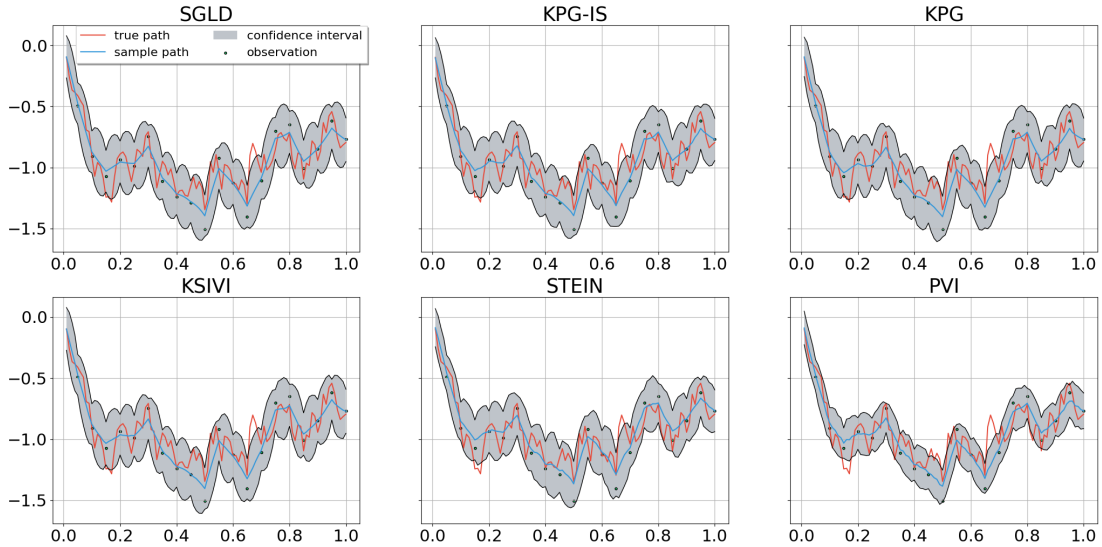

Figure 4: Comparison of posterior quality of different models (facets) for the diffusion process.

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 $\mathbb{E}\left[\hat{s}_{\mathrm{STEIN},k}(\boldsymbol{z})\right] = \mathbb{E}\left[\hat{s}_{\mathrm{SI},k}(\boldsymbol{z})\right]$ it follows that

$$\begin{aligned}
\Delta\mathbb{V} &:= \operatorname{trace}\left(\mathbb{V}\left[\hat{s}_{\mathrm{STEIN},k}(\boldsymbol{z})\right] - \mathbb{V}\left[\hat{s}_{\mathrm{SI},k}(\boldsymbol{z})\right]\right) && (32)\\
&= \operatorname{trace}\left(\mathbb{E}\left[\hat{s}_{\mathrm{STEIN},k}(\boldsymbol{z})\hat{s}_{\mathrm{STEIN},k}(\boldsymbol{z})^\top\right] - \mathbb{E}\left[\hat{s}_{\mathrm{SI},k}(\boldsymbol{z})\hat{s}_{\mathrm{SI},k}(\boldsymbol{z})^\top\right]\right) && (33)\\
&= \mathbb{E}\left[\hat{s}_{\mathrm{STEIN},k}(\boldsymbol{z})^\top\hat{s}_{\mathrm{STEIN},k}(\boldsymbol{z})\right] - \mathbb{E}\left[\hat{s}_{\mathrm{SI},k}(\boldsymbol{z})^\top\hat{s}_{\mathrm{SI},k}(\boldsymbol{z})\right] && (34)\\
&= \mathbb{E}\left\|\hat{s}_{\mathrm{STEIN},k}(\boldsymbol{z})\right\|_2^2 - \mathbb{E}\left\|\hat{s}_{\mathrm{SI},k}(\boldsymbol{z})\right\|_2^2. && (35)
\end{aligned}$$

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

$$\begin{aligned}
\Delta\mathbb{V} &= \frac{1}{n}\mathbb{E}_{\boldsymbol{z}'\sim q_{\boldsymbol{z}}}\left[k(\boldsymbol{z},\boldsymbol{z}')^2\cdot\frac{\|\boldsymbol{z}'-\boldsymbol{z}\|^2}{\sigma_k^4}\right]\\
&\quad - \frac{1}{n}\mathbb{E}_{\boldsymbol{\epsilon}',\boldsymbol{\eta}'\sim p_{\boldsymbol{\epsilon},\boldsymbol{\eta}}}\left[k(\boldsymbol{z},\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}'+\boldsymbol{\mu}_{\boldsymbol{\epsilon}'})^2\cdot\|\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^{-1}\boldsymbol{\eta}'\|_2^2\right]. && (38)\\
&= \frac{1}{n}\mathbb{E}_{\boldsymbol{\epsilon}',\boldsymbol{\eta}'\sim p_{\boldsymbol{\epsilon},\boldsymbol{\eta}}}k(\boldsymbol{z},\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}'+\boldsymbol{\mu}_{\boldsymbol{\epsilon}'})^2\\
&\quad\cdot\left[\frac{\|\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}'+\boldsymbol{\mu}_{\boldsymbol{\epsilon}'}-\boldsymbol{z}\|_2^2}{\sigma_k^4} - \|\operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})^{-1}\boldsymbol{\eta}'\|_2^2\right]. && (39)
\end{aligned}$$

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

$$
\begin{aligned}
\Delta\mathbb{V} &= \mathbb{E}\left[\alpha\beta\right] && (45)\\
&\leq \mathbb{E}\left[\alpha\right] \cdot \mathbb{E}\left[\beta\right] && (46)\\
&= \frac{1}{n}\mathbb{E}_{\boldsymbol{\epsilon}', \boldsymbol{\eta}' \sim p_{\boldsymbol{\epsilon},\boldsymbol{\eta}}} k(\boldsymbol{z}, \mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'})\boldsymbol{\eta}' + \boldsymbol{\mu}_{\boldsymbol{\epsilon}'})^2 \\
&\quad \cdot \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}', \boldsymbol{\eta}' \sim p_{\boldsymbol{\epsilon},\boldsymbol{\eta}}} \left[\frac{\|\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'} \odot \boldsymbol{\eta}'\|_2^2 + 2(\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'} \odot \boldsymbol{\eta}')^\top(\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}) + \|\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2}{\sigma_k^4} - \|\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'}^{\odot -1} \odot \boldsymbol{\eta}'\|_2^2\right] && (47)\\
&\overset{\boldsymbol{\eta}\sim\mathcal{N}(0,I)}{=} \frac{1}{n}\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{z}}} k(\boldsymbol{z}, \boldsymbol{z}')^2 \\
&\quad \cdot \underbrace{\left(\frac{\mathbb{E}_{\boldsymbol{\epsilon}'\sim p_{\boldsymbol{\epsilon}}}\left[\|\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'}\|_2^2 + \|\boldsymbol{\mu}_{\boldsymbol{\epsilon}'} - \boldsymbol{z}\|_2^2\right]}{\sigma_k^4} - \mathbb{E}_{\boldsymbol{\epsilon}'\sim p_{\boldsymbol{\epsilon}}}\left[\|\boldsymbol{\sigma}_{\boldsymbol{\epsilon}'}^{\odot -1}\|_2^2\right]\right)}_{=:\gamma}. && (48)\\
&= \frac{1}{n}\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{z}}}\left[\frac{1}{(2\pi\sigma_k^2)^{d_z}} \exp\left(-\frac{\|\boldsymbol{z} - \boldsymbol{z}'\|_2^2}{\sigma_k^2}\right)\right] \cdot \gamma && (49)\\
&\approx \frac{q_{\boldsymbol{z}}(\boldsymbol{z})}{n(2\sqrt{\pi}\sigma_k)^{d_z}} \cdot \gamma. && (50)
\end{aligned}
$$

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 $\boldsymbol{\epsilon}$-samples. Reported are mean $\pm$ 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.

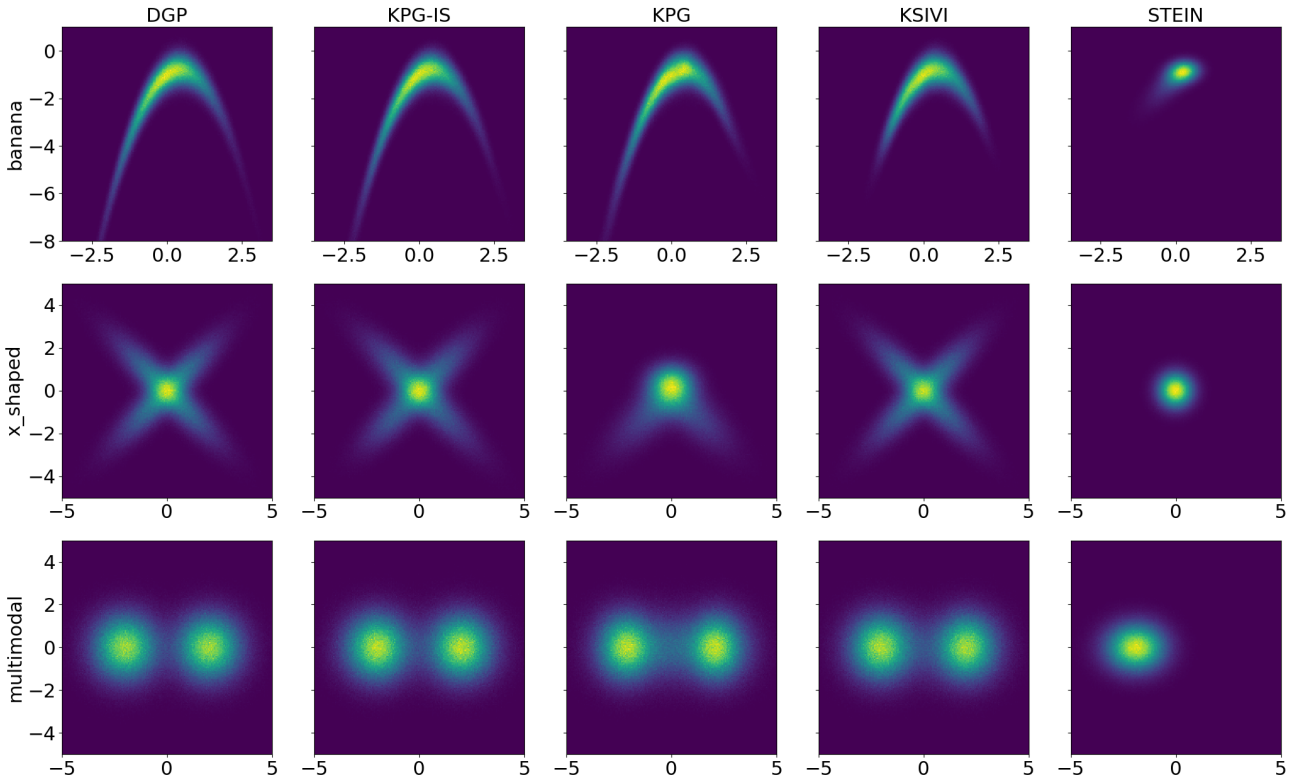

Figure 5: Density contour plots for the DGP and each method from one representative repetition. Each plot visualizes the estimated density using 100,000 $\boldsymbol{\epsilon}$-samples. The DGP serves as the reference distribution, while the others illustrate the approximation quality of each method.

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 $j = 1, \ldots, l$, 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.

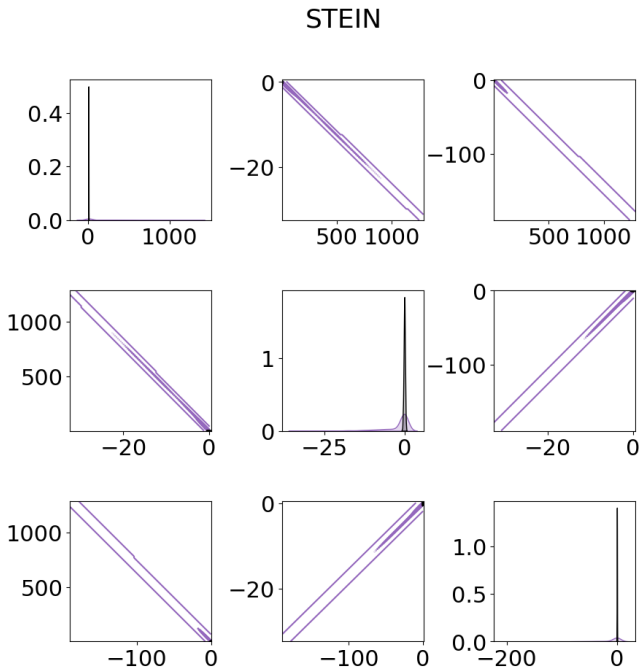

Figure 6: STEIN: Marginal and pairwise density estimates for $\boldsymbol{\beta}^{(1)}$, $\boldsymbol{\beta}^{(2)}$, and $\boldsymbol{\beta}^{(3)}$, with SGLD estimates shown in black for reference.

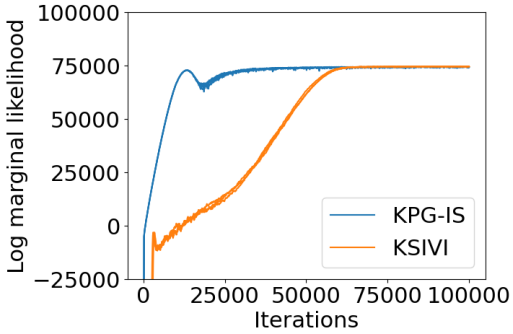

Figure 7: Convergence speed comparison between the current state-of-the-art method KSIVI and our proposal KPG-IS based on 3 repetitions.