Sparse Inducing Points in Deep Gaussian Processes: Enhancing Modeling with Denoising Diffusion Variational Inference

Jian Xu [1] **Delu Zeng** [1] **John Paisley** [2]

Abstract

Deep Gaussian processes (DGPs) provide a robust paradigm for Bayesian deep learning. In DGPs, a set of sparse integration locations called inducing points are selected to approximate the posterior distribution of the model. This is done to reduce computational complexity and improve model efficiency. However, inferring the posterior distribution of inducing points is not straightforward. Traditional variational inference approaches to posterior approximation often lead to significant bias. To address this issue, we propose an alternative method called Denoising Diffusion Variational Inference (DDVI) that uses a denoising diffusion stochastic differential equation (SDE) to generate posterior samples of inducing variables. We rely on score matching methods for denoising diffusion model to approximate score functions with a neural network. Furthermore, by combining classical mathematical theory of SDEs with the minimization of KL divergence between the approximate and true processes, we propose a novel explicit variational lower bound for the marginal likelihood function of DGP. Through experiments on various datasets and comparisons with baseline methods, we empirically demonstrate the effectiveness of DDVI for posterior inference of inducing points for DGP models.

1. Introduction

Deep Gaussian Processes (DGPs) (Damianou & Lawrence, 2013) have emerged as a robust framework for Bayesian deep learning (Fortuin, 2022) that allows for flexible modeling of complex functions. DGPs extend the idea of Gaussian Processes (GPs) (Rasmussen, 2003) to multiple layers, enabling the modeling of hierarchical structures and capturing intricate dependencies within the data. A crucial aspect of DGPs is the selection of inducing variables (Titsias, 2009; Snelson & Gharahmani, 2005; Quiñonero-Candela & Rasmussen, 2005), which are sparse integration locations used to approximate the posterior distribution of the model. By leveraging these inducing points, DGPs can efficiently handle large datasets and reduce the computational burden.

Variational inference methods (Blei et al., 2017; Zhang et al., 2018) aim to approximate the true posterior distribution with a parameterized variational distribution by minimizing their KL divergence. In the context of DGPs, traditional variational methods include mean-field Gaussian variational inference (DSVI) (Salimbeni & Deisenroth, 2017) and Implicit Posterior Variational Inference (IPVI) (Yu et al., 2019). However, both of these methods have their limitations and can introduce significant bias when learning the posterior distribution of inducing points.

DSVI approximates the posterior distribution of inducing points with a simple Gaussian distribution. Although this approximation is analytically tractable, it often leads to substantial bias when dealing with nonlinear likelihood functions. The simplifying assumptions made in the mean-field approximation can fail to capture the complex dependencies and correlations between the inducing points, resulting in suboptimal results. On the other hand, IPVI uses a neural network to parameterize the posterior distribution of inducing points. Posterior inference is formulated as a Nash equilibrium (Awerbuch et al., 2008) similar to that of generative adversarial networks (GANs) (Goodfellow et al., 2014), requiring adversarial learning for the max-max problem. However, optimizing this objective function can be challenging, especially when dealing with non-convex neural networks, and lead to instability during training and contribute to significant bias in the posterior inference of inducing points (Jenni & Favaro, 2019).

These limitations of traditional variational methods for inference of DGPs inspires the exploration of alternative approaches. Motivated by the success of denoising diffusion models (Rombach et al., 2022) in deep learning, we propose a Denoising Diffusion Variational Inference (DDVI) method that utilizes the denoising diffusion SDE and incorporates

[1]South China University of Technology, Guangzhou, China [2]Columbia University, New York, USA. Correspondence to: Delu Zeng <dlzeng@scut.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

principles similar to the score matching method (Song et al., 2020) in order to construct the objective function.

By employing the denoising diffusion SDE, we can accurately capture the complex dependencies and correlations among the inducing points. Additionally, similar to the score matching method, we can approximate the intricate score functions required for accurate posterior inference using a neural network. This combination finally allows us to explicitly derive a variational lower bound for the marginal likelihood function by KL divergence minimization, thereby addressing the bias introduced by traditional variational methods. Furthermore, DDVI incorporates numerous unique insights, including the well-developed mathematical theory of SDEs (Anderson, 1982; Haussmann & Pardoux, 1986), the bridge process trick, stochastic optimization techniques, reparameterization techniques, and gradient backpropagation. These collectively enable us to efficiently obtain posterior samples from the denoising diffusion network. As a result, our approach improves not only the computational efficiency but also ensures stable and reliable training in DGPs.

In summary, our contributions can be outlined as follows:

- We propose a novel parameterization approach for the posterior distribution of inducing points in DGPs, utilizing a denoising diffusion process. This method not only guarantees model efficiency by accurately capturing the complex dependencies and correlations among the inducing points, but also facilitates optimization and training.
- We exploit the minimization of KL divergence between the approximate and true processes to derive an explicit variational lower bound. To efficiently obtain posterior samples, we employ stochastic optimization and reparameterization techniques for gradient backpropagation within the denoising diffusion network.
- Through extensive experiments on various datasets and comparisons with baseline methods, we empirically demonstrate the effectiveness of the DDVI method in posterior inference of inducing points for DGP models.

2. Method

2.1. Model Review

2.1.1. Gaussian Process

Consider a random function $f : \mathbb{R}^D \to \mathbb{R}$ that maps N training inputs $\mathbf{X} \triangleq \{\mathbf{x}_n\}_{n=1}^N$ to a set of noisy observed outputs $\mathbf{y} \triangleq \{y_n\}_{n=1}^N$. Often, a zero mean Gaussian Process (GP) prior is assumed for the function, $f \sim \mathcal{GP}(0, k)$, where k denotes the covariance kernel function $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$. Let $\mathbf{f} \triangleq (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^\top$ represent the latent function values at the inputs $\mathbf{X}$. The GP prior assumption then induces a multivariate Gaussian prior over the function values, expressed as $p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K_{XX}})$, where the covariance matrix $\mathbf{K_{XX}}$ is defined by $[\mathbf{K_{XX}}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. The observed outputs $\mathbf{y}$ are then assumed to be contaminated by i.i.d. noise, modeled as $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I})$, where σ^2 is the noise variance. The GP posterior distribution of the latent output $p(\mathbf{f}|\mathbf{y})$ has a closed-form solution. However, the computational cost is $\mathcal{O}(N^3)$ and the storage requirement is $\mathcal{O}(N^2)$, making it challenging to scale to large datasets without introducing additional techniques.

Sparse methods have been developed that introduce *inducing points* $\mathbf{Z} = \{\mathbf{z}_m\}_{m=1}^M$ from the input space, along with corresponding *inducing variables*: $\mathbf{u} = \{f(\mathbf{z}_m)\}_{m=1}^M$. These methods reduce the computational complexity to $\mathcal{O}(NM^2)$. In the *Sparse Gaussian Processes* (SGPs) framework, the inducing variables $\mathbf{u}$ and the function values $\mathbf{f}$ share a joint multivariate Gaussian distribution, expressed as $p(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u})p(\mathbf{u})$, with the conditional distribution given by

$$p(\mathbf{f}|\mathbf{u}) = \mathcal{N}(\mathbf{f}|\mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{u}, \mathbf{K_{XX}} - \mathbf{K_{XZ}}\mathbf{K_{ZZ}^{-1}}\mathbf{K_{ZX}}) \quad (1)$$

and $p(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{K_{ZZ}})$ is the prior over the outputs of the inducing points.

2.1.2. Deep Gaussian Processes

A multi-layer Deep Gaussian Process (DGP) model is a hierarchical composition of Gaussian Process (GP) models constructed by stacking multiple-output Sparse GPs (SGPs) together, as described in (Damianou & Lawrence, 2013). Consider a DGP model with L layers, where each layer $\ell = 1, \ldots, L$ consists of D_ℓ independent random functions. The output of the $(\ell-1)^{\text{th}}$ layer, denoted as $\mathbf{F}_{\ell-1}$, serves as the input to the ℓ^{th} layer. Formally, the outputs of the ℓ^{th} layer are defined as $\mathbf{F}_\ell \triangleq \{f_{\ell,1}(\mathbf{F}_{\ell-1}), \cdots, f_{\ell,D_\ell}(\mathbf{F}_{\ell-1})\}$, where $f_{\ell,d} \sim \mathcal{GP}(0, k_\ell)$ for $d = 1, \ldots, D_\ell$, and $\mathbf{F}_0 \triangleq \mathbf{X}$. The inducing points and their corresponding inducing variables for each layer are denoted by $\boldsymbol{\mathcal{Z}} \triangleq \{\mathbf{Z}_\ell\}_{\ell=1}^L$ and $\mathcal{U} \triangleq \{\mathbf{U}_\ell\}_{\ell=1}^L$, respectively. Here, $\mathbf{U}_\ell \triangleq \{f_{\ell,1}(\mathbf{Z}_\ell), \cdots, f_{\ell,D_\ell}(\mathbf{Z}_\ell)\}$. Let $\mathcal{F} \triangleq \{\mathbf{F}_\ell\}_{\ell=1}^L$. The design of the DGP model leads to the following joint model density,

$$p(\mathbf{y}, \mathbf{F}, \mathbf{U}) = p(\mathbf{y}|\mathbf{F}_L) \prod_{\ell=1}^{L} p(\mathbf{F}_\ell|\mathbf{F}_{\ell-1}, \mathbf{U}_\ell) p(\mathbf{U}). \quad (2)$$

Here we place independent GP priors within and across layers on $\mathbf{U}$,

$$p(\mathbf{U}) = \prod_{l=1}^{L} p(\mathbf{U}_l) = \prod_{l=1}^{L} \prod_{d=1}^{D} \mathcal{N}(\mathbf{U}_{\ell,d}|0, \mathbf{K_{Z_\ell Z_\ell}}) \quad (3)$$

and the condition similar to Eq. (1) is defined as follows,

$$p\left(\mathbf{F}_{\ell} \mid \mathbf{F}_{\ell-1}, \mathbf{U}_{\ell}\right)=\prod_{d=1}^{D_{\ell}} \mathcal{N}\left(\mathbf{F}_{\ell, d} \mid \mu_{\ell, d}, \mathbf{\Sigma}_{\ell, \mathbf{d}}\right), \quad (4)$$

where we define

$$\begin{aligned}\mu_{\ell,\mathbf{d}} &= \mathbf{K}_{\mathbf{F}_{\ell-1}\mathbf{Z}_{\ell}}\mathbf{K}^{-1}_{\mathbf{Z}_{\ell}\mathbf{Z}_{\ell}}\mathbf{U}_{\ell,d}\\ \mathbf{\Sigma}_{\ell,\mathbf{d}} &= \mathbf{K}_{\mathbf{F}_{\ell-1}\mathbf{F}_{\ell-1}}-\mathbf{K}_{\mathbf{F}_{\ell-1}\mathbf{z}_{\ell}}\mathbf{K}^{-1}_{\mathbf{Z}_{\ell}\mathbf{Z}_{\ell}}\mathbf{K}_{\mathbf{Z}_{\ell}\mathbf{F}_{\ell-1}}.\end{aligned}$$

In the context of DGPs, traditional variational methods primarily include mean-field Gaussian variational inference (DSVI) (Salimbeni & Deisenroth, 2017) and Implicit Posterior Variational Inference (IPVI) (Yu et al., 2019). However, both of these methods have their limitations and can introduce significant bias in inferring the posterior distribution of inducing points.

DSVI approximates the posterior by a mean-field Gaussian, $q\left(\mathbf{U}_{\ell,1:D_{\ell}}\right)=\mathcal{N}\left(\mathbf{U}_{\ell,1:D_{\ell}} \mid \mathbf{m}_{\ell,1:D_{\ell}}, \mathbf{S}_{\ell,1:D_{\ell}}\right)$, where $\mathbf{m}_{\ell,1:D_{\ell}}$ and $\mathbf{S}_{\ell,1:D_{\ell}}$ are variational parameters. However, this assumption is overly strict and may limit the effectiveness and expressiveness of the model. The likelihood $p(\mathbf{y}|\mathbf{U})$ is difficult to compute because the latent functions $\mathbf{F}_1, \cdots, \mathbf{F}_{L-1}$ are all non-linear kernel functions.

On the other hand, IPVI utilizes a neural network ϕ to parameterize the posterior distribution of inducing points. Posterior inference is formulated as a Nash equilibrium (Awerbuch et al., 2008) similar to that of generative adversarial networks (GANs) (Goodfellow et al., 2014), requiring adversarial learning for the max-max problem,

$$\begin{aligned} & l_{\mathrm{IPVI}}(\xi)=\mathbb{E}_{q_{\xi}(\mathbf{U})}\left[\log p(\mathbf{y}|\mathbf{U})-D_{\psi^{\star}}(\mathbf{U})\right] \\ & s.t. \quad \psi^{\star}=\arg\max_{\psi} \mathbb{E}_{p(\mathbf{U})}[\log\left(1-\sigma\left(D_{\psi}(\mathbf{U})\right)\right] \\ & \qquad +\mathbb{E}_{q_{\xi}(\mathbf{U})}[\log\sigma\left(D_{\psi}(\mathbf{U})\right)], \end{aligned} \quad (5)$$

where D_{ψ} is another discriminator network. However, optimizing such an implict objective $l(\xi)$ can be challenging, especially when dealing with the non-convex neural network D_{Ψ}. This can lead to instability during training and contribute to significant bias in the posterior inference of inducing points (Jenni & Favaro, 2019).

To address this issue, we propose a novel parameterization approach for the posterior distribution of inducing variables $\mathbf{U}$ that uses a denoising diffusion process. This method not only ensures model efficiency by accurately capturing the complex dependencies and correlations among the inducing points, but also facilitates optimization and training.

2.2. Denoising Diffusion Varational Inference

2.2.1. Parameterizing Inducing Point Posteriors

Let $H=D\times M\times L$ denote the dimension of the inducing points. We aim to sample from the true posterior distribution $q(\mathbf{U})$ in $\mathbb{R}^H$, $q(\mathbf{U})=p(\mathbf{U}|\mathbf{y})$. Following a similar setup to prior works (Tzen & Raginsky, 2019; Zhang & Chen, 2021; Vargas et al., 2023), we start by sampling from a fixed distribution p_{fix} and then follow a Markov process in which we consider a sequential latent variable model with a joint distribution denoted as $\mathcal{Q}\left(\mathbf{U}_0,\ldots,\mathbf{U}_T\right)$, for step $t_s\in\{0,..,T-1\}$,

$$\mathbf{U}_{t_s+1}\sim\mathcal{T}\left(\mathbf{U}_{t_s+1}\mid\mathbf{U}_{t_s}\right), \qquad \mathbf{U}_0\sim p_{\text{fix}} \quad (6)$$

Here $\mathcal{T}\left(\mathbf{U}_{t_s+1}\mid\mathbf{U}_{t_s}\right)$ denotes a transition probability distribution. Through this sequence model, we use the marginal distribution $\mathcal{Q}\left(\mathbf{U}_T\right)$ at the terminal step T to approximate the true posterior distribution $q\left(\mathbf{U}_T\right)$.

2.2.2. Time-reversal Representation of Diffusion SDE

In this paper, we constrain the Markov process $\mathcal{Q}\left(\mathbf{U}_0,\ldots,\mathbf{U}_T\right)$ to be a time-reversal process of the following forward noising diffusion stochastic differential equation (SDE),

$$\mathrm{d}\overrightarrow{\mathbf{U}}_t=\mathbf{h}(t,\overrightarrow{\mathbf{U}}_t)\mathrm{d}t+g(t)\mathrm{d}B_t, \qquad \overrightarrow{\mathbf{U}}_0\sim q, \quad (7)$$

where $\mathbf{h}(t,\cdot):\mathbb{R}^H\to\mathbb{R}^H$ is the drift coefficient, $g(t)\in\mathbb{R}$ is the diffusion coefficient, and $(B_t)_{t\in[0,T]}$ is an H-dimensional Brownian motion. This diffusion induces the path measure $\mathcal{P}$ on the time interval $[0,T]$ and the marginal density of $\overrightarrow{\mathbf{U}}_t$ is denoted p_t. Note that by definition we always have $p_0=q$ when using an SDE to perturb this distribution. In DDPM (Ho et al., 2020; Song et al., 2020), p_T is an unstructured prior distribution that contains no information of p_0, such as a Gaussian distribution with fixed mean and variance.

From (Anderson, 1982; Haussmann & Pardoux, 1986), the time-reversal representation of Eq. (7), $\overleftarrow{\mathbf{U}}_t=\overrightarrow{\mathbf{U}}_{T-t}$, where equality is here in distribution, satisfies

$$\begin{aligned}\mathrm{d}\overleftarrow{\mathbf{U}}_t &= g(T-t)^2\nabla\ln\left(p_{T-t}(\overleftarrow{\mathbf{U}}_t)\right)\mathrm{d}t-\mathbf{h}(T-t,\overleftarrow{\mathbf{U}}_t)\,\mathrm{d}t\\ &\quad +g(T-t)\,\mathrm{d}W_t,\\ \overleftarrow{\mathbf{U}}_0 &\sim p_T,\end{aligned} \quad (8)$$

where $(W_t)_{t\in[0,T]}$ is another H-dimensional Brownian motion. By definition, this time-reversal starts from

$$\overleftarrow{\mathbf{U}}_0\sim p_T\approx p_{\text{fix}}$$

and is such that $\overleftarrow{\mathbf{U}}_T\sim q$. Since the distribution of $\overleftarrow{\mathbf{U}}_T$ is consistent with the true posterior q , we can parameterize the transition probability $\mathcal{T}\left(\mathbf{U}_{t_s+1}\mid\mathbf{U}_{t_s}\right)$ in the Euler discretized form of Eq. (8).

2.2.3. SCORE MATCHING TECHNIQUE

This suggests that if we could approximately simulate the diffusion of Eq. (8), then we could obtain approximate samples from the target q. However, putting this idea in practice requires being able to approximate the intractable scores $\nabla \ln \left(p_t(\cdot)\right)$ for $t \in [0, T]$. To achieve this, DDPM (Ho et al., 2020; Song et al., 2020) rely on score matching techniques. Specially, to approximate $\mathcal{P}$ consider a path measure $\mathcal{P}^\phi$ whose time-reversal is induced by

$$\begin{aligned} \mathrm{d}\overleftarrow{\mathbf{U}}_t^\phi &= g(T-t)^2 s_\phi(T-t, \overleftarrow{\mathbf{U}}_t^\phi)\mathrm{d}t - \mathbf{h}(T-t, \overleftarrow{\mathbf{U}}_t^\phi)\mathrm{d}t \\ &\quad + g(T-t)\,\mathrm{d}W_t, \\ \overleftarrow{\mathbf{U}}_0^\phi &\sim p_{\mathrm{fix}}, \end{aligned} \tag{9}$$

so that the backward process $\overleftarrow{\mathbf{U}}_t^\phi \sim \mathcal{Q}_t^\phi$, where p_{fix} represents a fixed distribution. To obtain $s_\phi(t, \cdot) \approx \nabla \ln \left(p_t(\cdot)\right)$, we parameterize $s_\theta(t, \cdot)$ by a neural network whose parameters are obtained by minimizing $\mathrm{KL}(\mathcal{P}||\mathcal{P}^\phi)$. From the chain rule for the KL divergence (Léonard, 2013) we have,

$$\mathrm{KL}(\mathcal{P}||\mathcal{P}^\phi) = \mathrm{KL}(p_T||p_{\mathrm{fix}}) + \mathrm{KL}(\mathcal{P}\cdot|\overrightarrow{\mathbf{U}}_T)||\mathcal{P}^\phi(\cdot|\overrightarrow{\mathbf{U}}_T^\phi)) \tag{10}$$

where by the well-known Girsanov Theorem (Oksendal, 2013) and the martingale property of Itô integrals the second term on the RHS is

$$\mathrm{KL}(\mathcal{P}(\cdot|\overrightarrow{\mathbf{U}}_T)||\mathcal{P}^\phi(\cdot|\overrightarrow{\mathbf{U}}_T^\phi)) = \tag{11}$$

$$\frac{1}{2}\int_0^T \mathbb{E}_{\mathcal{P}_t}\left[g(t)^2 \left\|\nabla \ln \left(p_t(\overrightarrow{\mathbf{U}}_t)\right) - \boldsymbol{s_\phi}(\overrightarrow{\mathbf{U}}_t, t)\right\|_2^2\right] \mathrm{d}t$$

From the denoising score matching derivation (Vincent, 2011), this can also be written as

$$\frac{1}{2}\int_0^T \mathbb{E}\left[g(t)^2 \|\nabla \ln \left(p_t(\overrightarrow{\mathbf{U}}_t|\overrightarrow{\mathbf{U}}_0)\right) - \boldsymbol{s_\phi}(\overrightarrow{\mathbf{U}}_t, t)\|_2^2\right] \mathrm{d}t$$

plus a constant term, where the expectation is over the joint distribution $p_0(\overrightarrow{\mathbf{U}}_0)p_t(\overrightarrow{\mathbf{U}}_t|\overrightarrow{\mathbf{U}}_0)$.

As the main loss function in DDPM, diffusion-based generative modeling approaches typically rely on Eq. (10), which involves sampling from p_0, the original data such as images, and then backpropagating to estimate the parameters of the neural network s_ϕ. However, unlike traditional score matching techniques, this loss function is not applicable to our model since our p_0 is the posterior probability q and we only have access to the joint likelihood, and cannot sample from it. To address this issue, we propose an alternative approach by minimizing $\mathrm{KL}(\mathcal{P}^\phi||\mathcal{P})$. Analogous to Eq. (10), considering that we can only obtain samples from $\mathcal{Q}_t^\phi$, we have,

$$\begin{aligned} \mathrm{KL}(\mathcal{P}^\phi||\mathcal{P}) &= \mathrm{KL}(\mathcal{Q}^\phi||\mathcal{Q}) \\ &= \mathrm{KL}(p_{\mathrm{fix}}||p_T) + \mathrm{KL}(\mathcal{Q}^\phi(\cdot|\overleftarrow{\mathbf{U}}_0^\phi)||\mathcal{Q}(\cdot|\overleftarrow{\mathbf{U}}_0)) \end{aligned} \tag{12}$$

where

$$\mathrm{KL}(\mathcal{Q}^\phi(\cdot|\overleftarrow{\mathbf{U}}_0^\phi)||\mathcal{Q}(\cdot|\overleftarrow{\mathbf{U}}_0)) = \frac{1}{2}\int_0^T \mathbb{E}\, g(T-t)^2 \|\varphi(t)\|_2^2 \mathrm{d}t \tag{13}$$

where the expectation is over $\mathcal{Q}_t^\phi$ and we have defined

$$\varphi(t) \triangleq \nabla \ln \left(p_{T-t}(\overleftarrow{\mathbf{U}}_t^\phi)\right) - \boldsymbol{s_\phi}(T-t, \overleftarrow{\mathbf{U}}_t^\phi)$$

However, the current challenge we face is that, although we can obtain samples from $\mathcal{Q}_t^\phi$ by simulating the SDE (9), dealing with the nonlinear drift function of SDE (9) makes it difficult to obtain $\nabla \ln \left(p_{T-t}(\overleftarrow{\mathbf{U}}_t^\phi)\right)$ in Eq. (13).

2.2.4. BRIDGE PROCESS TRICK

Therefore, we propose an alternative approach by constructing a bridge process $\mathcal{P}^{\mathrm{Bri}}$ to assist in measuring $\mathrm{KL}(\mathcal{P}^\phi||\mathcal{P})$. First, we observe that

$$\begin{aligned} \mathrm{KL}(\mathcal{P}^\phi||\mathcal{P}) &= \mathbb{E}_{\mathcal{P}^\phi} \log \frac{\mathrm{d}\mathcal{P}^\phi}{\mathrm{d}\mathcal{P}} \\ &= \mathbb{E}_{\mathcal{P}^\phi} \log \frac{\mathrm{d}\mathcal{P}^\phi}{\mathrm{d}\mathcal{P}^{\mathrm{Bri}}} + \mathbb{E}_{\mathcal{P}^\phi} \log \frac{\mathrm{d}\mathcal{P}^{\mathrm{Bri}}}{\mathrm{d}\mathcal{P}} \end{aligned} \tag{14}$$

where we represent the stochastic process KL with the Radon-Nikodym derivative. Given the specific form in Eq. (14), we define the bridge process $\mathcal{P}^{\mathrm{Bri}}$ to follow the diffusion formula as in Eq. (7), but initialized at $p_0^{\mathrm{Bri}}(\overrightarrow{\mathbf{U}}_0^{\mathrm{Bri}}) = p_{\mathrm{fix}}$ instead of q, which aligns with the distribution of $\overleftarrow{\mathbf{U}}_0$ in Eq. (9),

$$\mathrm{d}\overrightarrow{\mathbf{U}}_t^{\mathrm{Bri}} = \mathbf{h}(t, \overrightarrow{\mathbf{U}}_t^{\mathrm{Bri}})\mathrm{d}t + g(t)\mathrm{d}B_t, \quad \overrightarrow{\mathbf{U}}_0^{\mathrm{Bri}} \sim p_{\mathrm{fix}}. \tag{15}$$

We typically assume $\mathbf{h}(\cdot, t)$ is affine, $\mathbf{h}(x, t) = -\lambda(t)\, x$ and $p_{\mathrm{fix}} = \mathcal{N}(0, \sigma^2 I)$. Then the transition kernel $p_t(\overrightarrow{\mathbf{U}}_t^{\mathrm{Bri}}|\overrightarrow{\mathbf{U}}_0^{\mathrm{Bri}})$ is always a Gaussian distribution $\mathcal{N}(l_t, \Sigma_t)$, where the mean l_t and variance Σ_t are often known in closed-forms (Särkkä & Solin, 2019) by

$$\begin{aligned} \frac{\mathrm{d}l_t}{\mathrm{d}t} &= -\lambda(t) l_t, & l_0 &= 0 \\ \frac{\mathrm{d}\Sigma_t}{\mathrm{d}t} &= -2\lambda(t)\Sigma_t + g(t)^2 I, & \Sigma_0 &= \sigma^2 I \end{aligned} \tag{16}$$

By the calculations of ordinary differential equations (Hale & Lunel, 2013), we obtain the following general solution to Eq. (16),

$$\begin{aligned} l_t &= l_0 e^{-\int_0^t \lambda(s)\mathrm{d}s}, \\ \Sigma_t &= \left(\int_0^t g(r)^2 e^{\int_0^r \lambda(s)\mathrm{d}s}\mathrm{d}r I + \Sigma_0\right) e^{-\int_0^t \lambda(s)\mathrm{d}s} \end{aligned} \tag{17}$$

According to Eq. (17) we can derive from the Gaussian linear transformation principle that for any t, the distribution

p_t^{Bri} of $\overrightarrow{\mathbf{U}}_t^{\text{Bri}}$ is a zero-mean Gaussian distribution,

$$\begin{aligned} p_t^{\text{Bri}}(\overrightarrow{\mathbf{U}}_t^{\text{Bri}}) &= \int p_t(\overrightarrow{\mathbf{U}}_t^{\text{Bri}}|\overrightarrow{\mathbf{U}}_0^{\text{Bri}}) p_t(\overrightarrow{\mathbf{U}}_0^{\text{Bri}}) d\overrightarrow{\mathbf{U}}_0^{\text{Bri}} \\ &= \mathcal{N}(0, \kappa_t) \end{aligned} \tag{18}$$

where we have defined the variance

$$\kappa_t \triangleq (e^{-\int_0^t \lambda(s)\mathrm{d}s})^2 \Sigma_0 + \Sigma_t.$$

We can write the SDE equation for the reverse process $\mathcal{Q}^{\text{Bri}}$ of $\mathcal{P}^{\text{Bri}}$ as

$$\begin{aligned} \mathrm{d}\overleftarrow{\mathbf{U}}_t^{\text{Bri}} &= g(T-t)^2 \nabla \ln \left(p_{T-t}^{\text{Bri}}(\overleftarrow{\mathbf{U}}_t^{\text{Bri}})\right) \mathrm{d}t \\ &\quad - \mathbf{h}(T-t, \overleftarrow{\mathbf{U}}_t^{\text{Bri}}) \mathrm{d}t \\ &\quad + g(T-t)\, \mathrm{d}W_t, \\ \overleftarrow{\mathbf{U}}_0^{\text{Bri}} &\sim p_T^{\text{Bri}}. \end{aligned} \tag{19}$$

According to Eq. (18), we can obtain an analytical expression for the derivative of the log-likelihood function with respect to $\overleftarrow{\mathbf{U}}_t^{\text{Bri}}$,

$$\nabla \ln \left(p_{T-t}^{\text{Bri}}(\overleftarrow{\mathbf{U}}_t^{\text{Bri}})\right) = -\frac{\overleftarrow{\mathbf{U}}_t^{\text{Bri}}}{\kappa_{T-t}}. \tag{20}$$

Next we calculate the value of Eq. (14). For the first term, according to the chain rule for KL and Girsanov Theorem (Oksendal, 2013), incorporating Eqs. (9, 19, 20), we have

$$\mathbb{E}_{\mathcal{P}^\phi} \log \frac{\mathrm{d}\mathcal{P}^\phi}{\mathrm{d}\mathcal{P}^{\text{Bri}}} = \mathrm{KL}\left(\mathcal{P}^\phi \| \mathcal{P}^{\text{Bri}}\right) = \mathrm{KL}\left(\mathcal{Q}^\phi \| \mathcal{Q}^{\text{Bri}}\right)$$

which can be broken down into the sum of two terms,

$$\mathrm{KL}\left(p_{\text{fix}} \| p_T^{\text{Bri}}\right) + \mathrm{KL}(\mathcal{Q}^\phi(\cdot|\overleftarrow{\mathbf{U}}_0^\phi) || \mathcal{Q}(\cdot|\overleftarrow{\mathbf{U}}_0^{\text{Bri}})) \tag{21}$$

where

$$\begin{aligned} &\mathrm{KL}(\mathcal{Q}^\phi(\cdot|\overleftarrow{\mathbf{U}}_0^\phi) || \mathcal{Q}(\cdot|\overleftarrow{\mathbf{U}}_0^{\text{Bri}})) = \\ &\frac{1}{2}\int_0^T \mathbb{E}_{\mathcal{Q}_t^\phi}\, g(T-t)^2 \left\| \frac{\overleftarrow{\mathbf{U}}_t^\phi}{\kappa_{T-t}} + \boldsymbol{s}_{\boldsymbol{\phi}}(T-t, \overleftarrow{\mathbf{U}}_t^\phi) \right\|_2^2 \mathrm{d}t \end{aligned}$$

At this point, we can simulate the SDE (9) to compute the first term in Eq. (14). The integral term can be computed using either ODE solvers (Chen et al., 2018) or by employing Riemann summation methods. For the second term, $\mathbb{E}_{\mathcal{P}^\phi} \log \frac{\mathrm{d}\mathcal{P}^{\text{Bri}}}{\mathrm{d}\mathcal{P}}$, we can see from Eq. (7) and Eq. (15) that $\mathcal{P}$ and $\mathcal{P}^{\text{Bri}}$ have the same dynamic system τ, except for different initial values. Therefore, we have

$$\begin{aligned} \mathbb{E}_{\mathcal{P}^\phi} \log \frac{\mathrm{d}\mathcal{P}^{\text{Bri}}}{\mathrm{d}\mathcal{P}} &= \mathbb{E}_{\mathcal{P}^\phi} \log \frac{\mathcal{P}^{\text{Bri}}(\tau|\cdot)\, p_0^{\text{Bri}}(\cdot)}{\mathcal{P}(\tau|\cdot)\, p_0(\cdot)} \\ &= \mathbb{E}_{\mathcal{Q}_T^\phi} \log \frac{p_0^{\text{Bri}}(\cdot)}{p_0(\cdot)} \\ &= \mathbb{E}_{\mathcal{Q}_T^\phi} \log \frac{p_{\text{fix}}}{q} \\ &= \mathbb{E}_{\mathcal{Q}_T^\phi} \log \frac{p_{\text{fix}}}{p(\mathbf{y}|\cdot)p(\cdot)} + \log p(\mathbf{y}) \end{aligned} \tag{22}$$

2.2.5. A New Evidence Lower Bound

Let $l_1(\phi) = \mathbb{E}_{\mathcal{P}^\phi} \log \frac{\mathrm{d}\mathcal{P}^\phi}{\mathrm{d}\mathcal{P}^{\text{Bri}}}$. Combining Eqs. (2, 14, 21, 22), we obtain a new variational lower bound $l(\phi)$ for the marginal likelihood $\log p(\mathbf{y})$ of our method,

$$\begin{aligned} \log p(\mathbf{y}) &= \mathrm{KL}(\mathcal{P}^\phi || \mathcal{P}) - l_1(\phi) - \mathbb{E}_{\mathcal{Q}_T^\phi} \log \frac{p_{\text{fix}}}{p(\mathbf{y}|\cdot)p(\cdot)} \\ &= \mathrm{KL}(\mathcal{P}^\phi || \mathcal{P}) - l_1(\phi) - \mathbb{E}_{\mathcal{Q}_T^\phi} \log p_{\text{fix}} \\ &\quad + \mathbb{E}_{\mathcal{Q}_T^\phi} \log p(\cdot) + \mathbb{E}_{\mathcal{Q}_T^\phi, \mathbf{F_1}, \ldots, \mathbf{F_L}} \log p(\mathbf{y}|\mathbf{F_L}) \\ &\geqslant \mathbb{E}_{\mathcal{Q}_T^\phi} \log p(\cdot) + \mathbb{E}_{\mathcal{Q}_T^\phi, \mathbf{F_1}, \ldots, \mathbf{F_L}} \log p(\mathbf{y}|\mathbf{F_L}) \\ &\quad - l_1(\phi) - \mathbb{E}_{\mathcal{Q}_T^\phi} \log p_{\text{fix}} \\ &= l(\phi) \end{aligned} \tag{23}$$

In our derivation, $p(\cdot)$ represents the prior function of $\mathbf{U}$. By introducing a new variational lower bound for $\log p(\mathbf{y})$, our proposed model, compared to the initial mean-field variational inference model (DSVI) where q is approximated by a Gaussian distribution, approximates the posterior distribution through a denoising diffusion process. The flexibility of the denoising neural network ϕ intuitively suggests that our model has an advantage in approximating the posterior distribution. On the other hand, compared to IPVI, DDVI provides an explicit evidence lower bound (ELBO), which means it is easier to train and allows for efficient backpropagation.

2.3. Reparameterization Trick and SGD

For ease of sampling, we consider a reparameterization version of Eq. (23) based on the approaximate transition probability $\mathcal{T}_\phi\left(\mathbf{U}_{t_s+1} \mid \mathbf{U}_{t_s}\right)$ given by

$$\begin{aligned} \mathcal{T}_\phi(\mathbf{U}_{t_s+1}) = \; & \mathbf{U}_{t_s} - \mathbf{h}(\mathbf{U}_{t_s}, T-t_s) + \\ & g(T-t_s)^2 s_\phi\left(T-t_s, \mathbf{U}_{t_s}\right) + g(T-t)\boldsymbol{\epsilon}_{t_s} \end{aligned} \tag{24}$$

where $\boldsymbol{\epsilon}_{t_s} \sim \mathcal{N}(0, I)$. Given that $\mathbf{U}_{t_s+1} = \mathcal{T}_\phi(\mathbf{U}_{t_s})$, we have a representation of $\mathbf{U}_{t_s}$ by a stochastic flow,

$$\mathbf{U}_{t_s+1} = \mathcal{T}_\phi\left(\mathbf{U}_{t_s}\right) = \mathcal{T}_\phi \circ \mathcal{T}_\phi \circ \cdots \mathcal{T}_\phi(\mathbf{U}_0). \tag{25}$$

Moreover, for DGP models, we also have a reparameterization version (Salimbeni & Deisenroth, 2017) of the conditional distribution in Eq. (4) of the form

$$\begin{aligned} \mathbf{F}_{\ell,d} = \; & \mathbf{K}_{\mathbf{F}_{\ell-1}\mathbf{Z}_\ell} \mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}^{-1} \mathbf{U}_{\ell,d} \\ & + \sqrt{\mathbf{K}_{\mathbf{F}_{\ell-1}\mathbf{F}_{\ell-1}} - \mathbf{K}_{\mathbf{F}_{\ell-1}\mathbf{Z}_\ell} \mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}^{-1} \mathbf{K}_{\mathbf{Z}_\ell\mathbf{F}_{\ell-1}}}\, \boldsymbol{\epsilon}_{\ell,d} \end{aligned} \tag{26}$$

where $\boldsymbol{\epsilon}_{\ell,d} \in \mathbb{R}^N$ are standard Gaussian random variables. In order to accelerate training and sampling in our inference

Algorithm 1 Denoising Diffuison Variational Inference (DDVI) algorithm for DGPs

Input: training data $\mathbf{X}, \mathbf{y}$ mini-batch size B
Initialize diffusion coefficient h, g, all DGP hyperparameters γ, denoising diffusion network parameters ϕ and set $l_0 = 0$
repeat
  for $t_s = 0$ **to** $T-1$ **do**
    Draw $\boldsymbol{\epsilon}_{t_s} \sim \mathcal{N}(0, I)$ and set $\mathbf{U}_{t_s+1} = \mathbf{U}_{t_s} - \mathbf{h}(\mathbf{U}_{t_s}, T-t_s) + g(T-t_s)^2 s_\phi\left(T-t_s, \mathbf{U}_{t_s}\right) + g(T-t)\boldsymbol{\epsilon}_{t_s}$
    Compute $\kappa_{T-(t_s+1)}$ by Eq. (18) and set $l_{t_s+1} = l_{t_s} + g(T-(t_s+1))^2 \| \frac{\mathbf{U}_{t_s+1}}{\kappa_{T-(t_s+1)}} + \boldsymbol{s_\phi}(T-(t_s+1), \mathbf{U}_{t_s+1}) \|_2^2$
  end for
  Sample mini-batch indices $I \subset \{1, \ldots, N\}$ with $|I| = B$ and set $\{\mathbf{U}_{\ell,1}, ..., \mathbf{U}_{\ell,D}\}_{\ell=1}^{L} = \mathbf{U}_T$
  for $\ell = 1$ **to** L **do**
    Draw $\epsilon_{\ell,d} \sim \mathcal{N}(0, I)$ and calculate $\mathbf{F}_{\ell,d} = \mathbf{K}_{\mathbf{F}_{\ell-1}\mathbf{Z}_\ell}\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}^{-1}\mathbf{U}_{\ell,d} + \sqrt{\mathbf{K}_{\mathbf{F}_{\ell-1}\mathbf{F}_{\ell-1}} - \mathbf{K}_{\mathbf{F}_{\ell-1}\mathbf{z}_\ell}\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}^{-1}\mathbf{K}_{\mathbf{Z}_\ell\mathbf{F}_{\ell-1}}}\boldsymbol{\epsilon}_{\ell,d}$
  end for
  Set $l(\phi, \gamma) = -\log p_{\text{fix}}\left(\mathbf{U}_T\right) + \log p\left(\mathbf{U}_T\right) + \frac{N}{B}\log p\left(\mathbf{y}_I \mid \mathbf{F}_L\right) - \mathrm{KL}\left(p_{\text{fix}} \parallel \mathcal{N}(0, \kappa_T)\right) - \frac{1}{2} l_T$
  Make a gradient descent update of $l(\phi, \gamma)$
until ϕ, γ converge

scheme, we propose a scalable variational bound that is tractable in the large data regime based on stochastic variational inference (Kingma & Welling, 2013; Hoffman & Blei, 2015; Salimbeni & Deisenroth, 2017; Naesseth et al., 2020) and stochastic gradient descent (Welling & Teh, 2011; Chen et al., 2014; Zou et al., 2019; Alexos et al., 2022). Specially, instead of computing the full log likelihood, we use a stochastic variant to subsample datasets into a mini-batches $\mathcal{D}_I$ with $|\mathbf{X}_I| = B$, where $I \subset \{1, 2, .., N\}$ is the index of a mini-batch. We present the resulting stochastic inference for our Denoising Diffuison Variational Inference algorithm for DGP models in Algorithm 1.

2.4. Predictive Distribution

To obtain the final layer density for making predictions, we first sample from the optimized generator and transform the input locations $\mathbf{x}$ to the test locations $\mathbf{x}^\star$ using Eq. (2). We subsequently compute the function values at the test locations, which are represented as $\mathbf{F}_\ell^\star$. Finally, we use the equation below to estimate the density of the final layer, which enables us to make predictions for the test data

$$q(\mathbf{F}_L^\star) = \int \prod_{d,\ell} p(\mathbf{F}_{\ell,d}^\star | \mathbf{F}_{\ell-1}^\star, \mathbf{U}_{\ell,d}) \mathcal{Q}_T^\phi\left(\mathbf{U}_{\ell,d}\right) d\mathbf{F}_{\ell-1}^\star d\mathbf{U}_{\ell,d}$$

where $\mathcal{Q}_T^\phi$ represents the output of the denoising diffusion process at time T and the first term of the integral $p(\mathbf{F}_{\ell,d}^\star | \mathbf{F}_{\ell-1}^\star, \mathbf{U}_{\ell,d})$ is conditional Gaussian. We leverage this to draw samples from $q\left(\mathbf{F}_L^\star\right)$ and further perform the sampling according to the problem considered.

3. Experiments

3.1. Baseline Models and Hyperparameter Settings

In order to evaluate the performance of our proposed method, we conducted empirical evaluations on real-world datasets for both regression and classification tasks, with both small and large datasets. We compare against several other models, including Doubly Stochastic VI (DSVI) (Salimbeni & Deisenroth, 2017), Implicit Posterior VI (IPVI) (Yu et al., 2019), and the state-of-the-art SGHMC model (Havasi et al., 2018). All experiments were conducted with the same hyper-parameters and initializations whenever possible to obtain a fair comparison.

We constructed a random $0.9/0.1$ train/test split and normalized the features of our datasets to the range $[-1, 1]$. The depth L of DGP models varied from 2 to 5, with 128 inducing points per layer, which were initialized by sampling Gaussian random variables. The output dimension for each hidden layer is set to 1 for the final layer and the dimensionality of the data for all others. We use the RBF kernel for all tasks. For all datasets, we have optimized hyper-parameters and network parameters jointly and set learning rate to 0.01 using Adam optimizer (Kingma & Ba, 2014). We trained all models on all datasets until convergence was achieved. In each experiment, we repeated the process 10 times and reported the mean and standard deviation of the metrics. The selection of the denoising diffusion networks are done manually by the classical grid search approach for each experimental dataset. It is worth mentioning that this work also benefits from the contributions of the PyTorch platform, GPyTorch (Gardner et al., 2018), and related work on neural SDE solvers (Li et al., 2020; Kidger et al., 2021). All our experiments were conducted on an RTX 4090 GPU.

3.2. Regression Task

In our experiments, we evaluated the performance of the DDVI model on ten UCI regression datasets, which varied in size from 308 to 2,055,733 data points. We used the mean RMSE and mean NLL (Gneiting & Raftery, 2007) of the test data as the performance metric, and the results are

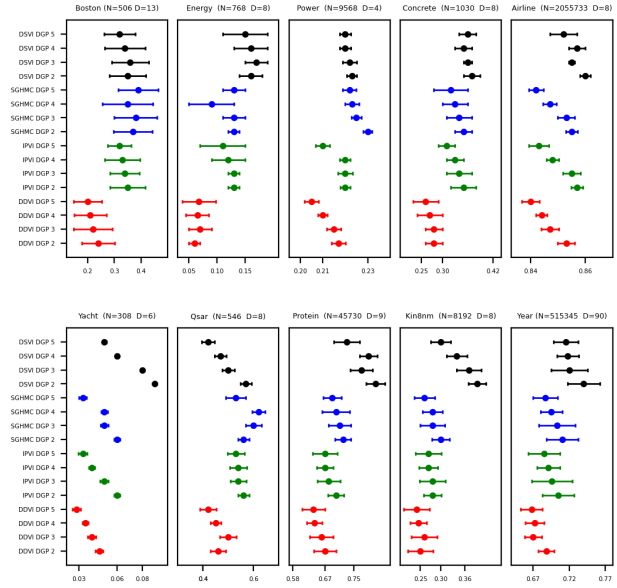

Figure 1. Regression test RMSE results by our DDVI method (red), SGHMC (blue), IPVI(green) and DSVI (black) for DGPs on ten UCI benchmark datasets. The numbers 2, 3, 4, and 5 represent the layers of DGP methods. Lower is better. The mean is shown with error bars of one standard error. The dimensions of the data are displayed above each subgraph.

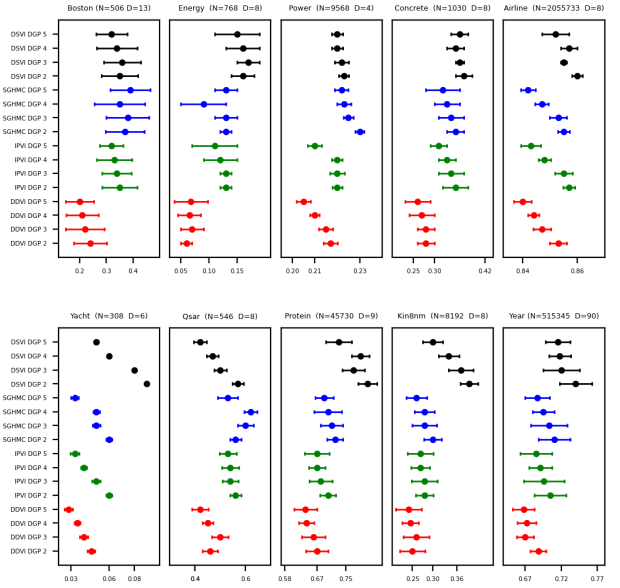

Figure 2. Regression test mean NLL results by our DDVI method (red), SGHMC (blue), IPVI (green) and DSVI (black) for DGPs on ten UCI benchmark datasets. The numbers 2, 3, 4, and 5 represent the layers of DGP methods. Lower is better. The mean is shown with error bars of one standard error. The dimensions of the data are displayed above each subgraph.

presented in Figure 1 and Figure 2.

As shown in these two figures, our DDVI method consistently achieves competitive results compared to three baselines on the majority of datasets. This is attributed to the key advantages of our approach, which overcomes limitations present in previous methods as discussed in the main text. Our findings also suggest that deeper DGP models tend to perform better. It is worth mentioning that the difference in performance may be attributed to the nature of the datasets, such as their size or the presence of outliers or singular values.

Using mini-batch algorithm and GPU acceleration, our method can also be extended to larger datasets. Our evaluation of the performance of DDVI in Figures 1 and 2 is also conducted on two real-world large-scale regression datasets: the YearMSD dataset and the Airline dataset. The YearMSD dataset has a large input dimension of $D = 90$ and a data size of approximately 500,000. The Airline dataset, on the other hand, has an input dimension of $D = 8$ and a large data size of approximately 2 million. For the YearMSD dataset, we split the data into training and test sets, using the first 463,810 examples as training data and the last 51,725 examples as test data. Similarly, for the Airline dataset, we take the first 700K points for training and next 100K for testing.

3.3. Image and Large-Scale Dataset Classification

We evaluate our method on multiclass classification tasks using the MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), and CIFAR-10 (Krizhevsky et al., 2009) datasets. The first two datasets consist of grayscale images of size 28×28 pixels, while CIFAR-10 comprises colored images of size 32×32 pixels. The results are presented in Table 1. We note that our method outperforms the other three methods on all three datasets, with significantly less training time. Specially, for CIFAR-10 dataset, we utilize the convolutional layers of ResNet-20 (He et al., 2016) as our feature extractor (Wilson et al., 2016) and achieve a remarkable accuracy of 95.56 on the test set. Additionally, we evaluate our approach using two large-scale classification datasets, the Higgs datase and the SUSY dataset, which are presented in Table 2.

3.4. Unsupervised Learning for Data Recovery Task

We conducted a reconstruction experiment on Frey Faces Data (Roweis & Saul, 2000), focusing on how models capture uncertainty when training with missing data in structured inputs. We used the entire dataset with a latent variable dimensionality of 20. The image data set contains 1965 images of a face taken from sequential frames of a short video.

Table 1. Mean test accuracy (%) and training details achieved by DSVI, SGHMC, IPVI and DDVI (ours) DGP model for three image classification datasets. Results are shown for 3 and 4 layers as indicated, and runtime is given per iteration.

Data Set	Model	Time3	Iter3	Acc3	Time4	Iter4	Acc4
MNIST	DSVI	0.34s	20K	97.17	0.54s	20K	97.41
	IPVI	0.49s	20K	97.58	0.62s	20K	97.80
	SGHMC	1.14s	20K	97.25	1.22s	20K	97.55
	DDVI	0.38s	20K	**98.84**	0.50s	20K	**99.01**
Fashion	DSVI	0.34s	20K	87.45	0.50s	20K	87.99
	IPVI	0.48s	20K	88.23	0.61s	20K	88.90
	SGHMC	1.21s	20K	86.88	1.25s	20K	87.08
	DDVI	0.40s	20K	**90.36**	0.55s	20K	**90.85**
CIFAR-10	DSVI	0.43s	20K	91.47	0.66s	20K	91.79
	IPVI	0.62s	20K	92.79	0.78s	20K	93.52
	SGHMC	8.04s	20K	92.62	8.61s	20K	92.94
	DDVI	0.45s	20K	**95.23**	0.69s	20K	**95.56**

Figure 3. The Brendan faces reconstruction task with 75% missing pixels. The top row represents the ground truth data and the bottom row showcases the reconstructions from the 20-dimensional latent distribution.

Table 2. Test AUC values for large-scale classification datasets. Uses random 90% / 10% training and test splits.

		SUSY	HIGGS
	N	5,500,000	11,000,000
	D	18	28
DSVI $M = 128$	$L = 2$	0.876	0.830
	$L = 3$	0.877	0.837
	$L = 4$	0.878	0.841
	$L = 5$	0.878	0.846
IPVI $M = 128$	$L = 2$	0.879	0.843
	$L = 3$	0.882	0.847
	$L = 4$	0.883	0.850
	$L = 5$	0.883	0.852
SGHMC $M = 128$	$L = 2$	0.879	0.842
	$L = 3$	0.881	0.846
	$L = 4$	0.883	0.850
	$L = 5$	0.884	0.853
DDVI $M = 128$	$L = 2$	**0.883**	**0.849**
	$L = 3$	**0.885**	**0.852**
	$L = 4$	**0.887**	**0.856**
	$L = 5$	**0.886**	**0.857**

Table 3. Mean RMSE and NLL achieved by DSVI, SGHMC, IPVI and DDVI (ours) GPLVM model for data recovery task. Standard deviation is shown in parentheses. Runtime is given per iteration.

Data Set	Model	Time	Iter	RMSE	NLL
Frey Faces	DSVI	0.32s	20K	8.32 (0.2)	1.49 (0.02)
	IPVI	0.42s	20K	7.91 (0.4)	1.33 (0.02)
	SGHMC	1.13s	20K	7.95 (0.3)	1.36 (0.03)
	DDVI	0.36s	20K	**7.64** (0.2)	**1.17** (0.01)

Each image is of size 20×28 yielding a 560 dimensional data space. In both cases, we chose 5% of the training set as missing data samples and removed 75% of their pixels, seeking to recover their original appearance. Figure 3 summarize the samples generated from the learned latent distribution. This reconstruction experiment is performed using the Gaussian Process Latent Variable Model (GPLVM) (Titsias & Lawrence, 2010) and is similar to the related work by (Gal et al., 2014).

To demonstrate the effectiveness of our method in producing more accurate likelihoods on image datasets, we present in Table 3 negative log-likelihood, and RMSE for reconstructed images on the Frey Faces, comparing with baseline methods. The results show that our method converges to higher likelihoods and lower RMSE, indicating superior performance in high-dimensional and multi-modal image data. This suggests that adding DDVI method can also improve the convergence of the traditional GPLVM methods.

4. Conclusion

We have introduced Denoising Diffusion Variational Inference (DDVI) as an alternative approach for inferring the posterior distribution of inducing points in Deep Gaussian Processes (DGPs). By employing a denoising diffusion stochastic differential equation (SDE) and utilizing the score matching method, we are able to accurately approximate challenging score functions using a neural network. Through extensive experiments and comparisons with baseline methods on various datasets, we demonstrated the effectiveness of DDVI in posterior inference of inducing points for DGP models. The DDVI method addressed the limitations of traditional variational inference techniques, reducing biases and improving accuracy in the posterior approximation. Our proposed DDVI approach not only enhances computational efficiency, but also provides a more robust framework for Bayesian deep learning with DGPs.

Acknowledgements

The work is supported by the Fundamental Research Program of Guangdong, China, under Grant 2023A1515011281; and in part by the National Natural Science Foundation of China under Grant 61571005.

Impact Statement

This paper aims to contribute to the advancement of the Machine Learning field. Our work may have various potential societal implications, none of which we believe need to be specifically emphasized here.

References

Alexos, A., Boyd, A. J., and Mandt, S. Structured stochastic gradient mcmc. In *International Conference on Machine Learning*, pp. 414–434. PMLR, 2022.

Anderson, B. D. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Awerbuch, B., Azar, Y., Epstein, A., Mirrokni, V. S., and Skopalik, A. Fast convergence to nearly optimal solutions in potential games. In *Proceedings of the 9th ACM conference on Electronic commerce*, pp. 264–273, 2008.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Chen, T., Fox, E., and Guestrin, C. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pp. 1683–1691. PMLR, 2014.

Cheng, C.-A. and Boots, B. Variational inference for gaussian process models with linear complexity. *Advances in Neural Information Processing Systems*, 30, 2017.

Damianou, A. and Lawrence, N. Deep Gaussian processes. In *Conference on Artificial Intelligence and Statistics*, pp. 207–215, 2013.

Fortuin, V. Priors in bayesian deep learning: A review. *International Statistical Review*, 90(3):563–591, 2022.

Gal, Y., Van Der Wilk, M., and Rasmussen, C. E. Distributed variational inference in sparse gaussian process regression and latent variable models. *Advances in neural information processing systems*, 27, 2014.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31, 2018.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Hale, J. K. and Lunel, S. M. V. *Introduction to functional differential equations*, volume 99. Springer Science & Business Media, 2013.

Haussmann, U. G. and Pardoux, E. Time reversal of diffusions. *The Annals of Probability*, pp. 1188–1205, 1986.

Havasi, M., Hernández-Lobato, J. M., and Murillo-Fuentes, J. J. Inference in deep Gaussian processes using stochastic gradient Hamiltonian Monte Carlo. In *Conference on Neural Information Processing Systems*, pp. 7517–7527, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proc. CVPR*, pp. 770–778, 2016.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Hoffman, M. D. and Blei, D. M. Structured stochastic variational inference. In *Artificial Intelligence and Statistics*, pp. 361–369, 2015.

Jenni, S. and Favaro, P. On stabilizing generative adversarial training with noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12145–12153, 2019.

Kidger, P., Foster, J., Li, X. C., and Lyons, T. Efficient and accurate gradients for neural sdes. *Advances in Neural Information Processing Systems*, 34:18747–18761, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2013.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images.(2009), 2009.

Lalchand, V. and Rasmussen, C. E. Approximate inference for fully bayesian gaussian process regression. In *Symposium on Advances in Approximate Bayesian Inference*, pp. 1–12. PMLR, 2020.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database, 1998. *URL http://www. research. att. com/˜ yann/ocr/mnist*, 1998.

Léonard, C. A survey of the schr\" odinger problem and some of its connections with optimal transport. *arXiv preprint arXiv:1308.0215*, 2013.

Li, X., Wong, T.-K. L., Chen, R. T., and Duvenaud, D. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pp. 3870–3882. PMLR, 2020.

Ma, C., Li, Y., and Hernández-Lobato, J. M. Variational implicit processes. In *International Conference on Machine Learning*, pp. 4222–4233. PMLR, 2019.

Mescheder, L., Nowozin, S., and Geiger, A. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *International conference on machine learning*, pp. 2391–2400. PMLR, 2017.

Naesseth, C., Lindsten, F., and Blei, D. Markovian score climbing: Variational inference with kl (p—— q). *Advances in Neural Information Processing Systems*, 33: 15499–15510, 2020.

Oksendal, B. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

Ortega, L. A., Santana, S. R., and Hernández-Lobato, D. Deep variational implicit processes. *arXiv preprint arXiv:2206.06720*, 2022.

Quiñonero-Candela, J. and Rasmussen, C. E. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

Rasmussen, C. E. Gaussian processes in machine learning. In *Summer school on machine learning*, pp. 63–71. Springer, 2003.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Rossi, S., Heinonen, M., Bonilla, E., Shen, Z., and Filippone, M. Sparse gaussian processes revisited: Bayesian approaches to inducing-variable approximations. In *International Conference on Artificial Intelligence and Statistics*, pp. 1837–1845. PMLR, 2021.

Roweis, S. T. and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500): 2323–2326, 2000.

Salimbeni, H. and Deisenroth, M. Doubly stochastic variational inference for deep Gaussian processes. In *Conference on Neural Information Processing Systems*, pp. 4588–4599, 2017.

Salimbeni, H., Cheng, C.-A., Boots, B., and Deisenroth, M. Orthogonally decoupled variational gaussian processes. *Advances in neural information processing systems*, 31, 2018.

Särkkä, S. and Solin, A. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

Shi, J., Titsias, M., and Mnih, A. Sparse orthogonal variational inference for gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 1932–1942. PMLR, 2020.

Snelson, E. L. and Gharahmani, Z. Sparse Gaussian processes using pseudo-inputs. In *Conference on Neural Information Processing Systems*, pp. 1257–1264, 2005.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Sun, S., Zhang, G., Shi, J., and Grosse, R. Functional variational bayesian neural networks. *arXiv preprint arXiv:1903.05779*, 2019.

Sun, S., Shi, J., Wilson, A. G., and Grosse, R. Scalable variational gaussian processes via harmonic kernel decomposition. *arXiv preprint arXiv:2106.05992*, 2021.

Titsias, M. and Lawrence, N. D. Bayesian gaussian process latent variable model. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 844–851. JMLR Workshop and Conference Proceedings, 2010.

Titsias, M. K. Variational learning of inducing variables in sparse Gaussian processes. In *Conference on Artificial Intelligence and Statistics*, pp. 567–574, 2009.

Tzen, B. and Raginsky, M. Theoretical guarantees for sampling and inference in generative models with latent diffusions. In *Conference on Learning Theory*, pp. 3084–3114. PMLR, 2019.

Vargas, F., Grathwohl, W., and Doucet, A. Denoising diffusion samplers. *arXiv preprint arXiv:2302.13834*, 2023.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Wilson, A. G., Hu, Z., Salakhutdinov, R. R., and Xing, E. P. Stochastic variational deep kernel learning. *Advances in neural information processing systems*, 29, 2016.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Yu, H., Chen, Y., Low, B. K. H., Jaillet, P., and Dai, Z. Implicit posterior variational inference for deep gaussian processes. *Advances in neural information processing systems*, 32, 2019.

Zhang, C., Bütepage, J., Kjellström, H., and Mandt, S. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.

Zhang, Q. and Chen, Y. Path integral sampler: a stochastic control approach for sampling. *arXiv preprint arXiv:2111.15141*, 2021.

Zou, D., Xu, P., and Gu, Q. Stochastic gradient hamiltonian monte carlo methods with recursive variance reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

A. Additional Related Works

A line of closely related work focusing on enhancing variational posteriors with decoupled/orthogonal inducing points has garnered significant attention in recent years (Cheng & Boots, 2017; Salimbeni et al., 2018; Shi et al., 2020; Sun et al., 2021). These methods present variational Gaussian process models that segregate the representation of mean and covariance functions within the reproducing kernel Hilbert space. This novel parametrization extends previous models and allows for solving the variational inference problem using stochastic gradient ascent with linear time and space complexity in the number of mean function parameters. In contrast to these approaches, our work diverges in its emphasis on precise posterior inference for Gaussian process models rather than complexity analysis concerning inducing points.

Another line of related work is on the idea of fully Bayesian Gaussian processes (Lalchand & Rasmussen, 2020; Rossi et al., 2021), particularly (Rossi et al., 2021) have applied MCMC-related methods and the ideas of fully Bayesian approaches to deep GPs, achieving significant improvements. Our work, on the other hand, focuses on improvements and advancements in the field of variational inference.

Additionally, another line of related work is on implict stochastic processes (Ma et al., 2019), that is not restricted to Gaussian predictive distributions, which is closely related to function-space models (Sun et al., 2019; Mescheder et al., 2017). Furthermore, (Ortega et al., 2022) further extends the idea of DGP models to implict stochastic processes, enhancing the flexibility of the models. We believe that this is a valuable complement to our approach, and future work may involve integrating our method with these advancements to explore more useful applications.