Neural Operator Variational Inference Based on Regularized Stein Discrepancy for Deep Gaussian Processes

Jian Xu, Shian Du, Junmei Yang, Qianli Ma, *Member, IEEE*, and Delu Zeng, *Member, IEEE*

***Abstract*— Deep Gaussian process (DGP) models offer a powerful nonparametric approach for Bayesian inference, but exact inference is typically intractable, motivating the use of various approximations. However, existing approaches, such as mean-field Gaussian assumptions, limit the expressiveness and efficacy of DGP models, while stochastic approximation can be computationally expensive. To tackle these challenges, we introduce neural operator variational inference (NOVI) for DGPs. NOVI uses a neural generator to obtain a sampler and minimizes the regularized Stein discrepancy (RSD) between the generated distribution and true posterior in $\mathcal{L}_2$ space. We solve the minimax problem using Monte Carlo estimation and subsampling stochastic optimization techniques and demonstrate that the bias introduced by our method can be controlled by multiplying the Fisher divergence with a constant, which leads to robust error control and ensures the stability and precision of the algorithm. Our experiments on datasets ranging from hundreds to millions demonstrate the effectiveness and the faster convergence rate of the proposed method. We achieve a classification accuracy of 93.56 on the CIFAR10 dataset, outperforming state-of-the-art (SOTA) Gaussian process (GP) methods. We are optimistic that NOVI possesses the potential to enhance the performance of deep Bayesian nonparametric models and could have significant implications for various practical applications.**

***Index Terms*— Deep Gaussian processes (DGPs), neural network generator, operator variational inference (VI).**

I. Introduction

GAUSSIAN processes (GPs) [1] are widely used in statistical inference and machine learning due to their effectiveness in modeling the relationship between inputs and outputs. For example, they have been successfully applied to modeling the dynamics of complex systems, such as robots or autonomous vehicles, for tasks such as trajectory planning [2], adaptive control [3], and anomaly detection [4]. The assumption that the latent function values follow a joint Gaussian distribution may not always hold, and in some scenarios, it can be overly restrictive [5]. For example, when dealing with non-Gaussian and nonstationary processes [6], [7], such as those found in financial time series or climate modeling, the Gaussian assumption may not be appropriate. Therefore, deep GPs (DGPs) have been proposed as an alternative approach to address these limitations in GP models.

A DGP model is a hierarchical composition of GP models that offers a probabilistic nonparametric approach with robust uncertainty quantification [8]. The non-Gaussian distribution over composition functions provides expressive capacity but also presents challenges for inference [9]. Previous research on DGP models has used variational inference (VI) with a combination of sparse GPs (SGPs) [10], [11], [12], [13] and mean-field Gaussian assumptions [14], [15], [16], [17], [18], [19], [20], [21] to approximate the posterior distribution. Stochastic optimization techniques have been used to scale up DGPs to handle large datasets, such as doubly stochastic VI (DSVI) [22].These strategies often incorporate a collection of inducing points ($M \ll N$) whose position is learned alongside the other model hyperparameters, reducing the training cost to $\mathcal{O}(NM^2)$.

The mean-field Gaussian assumptions in approximate posterior distributions simplify computations but can impose overly stringent constraints on DGP models, potentially limiting their expressiveness and effectiveness. Stochastic approximation approaches, such as SGHMC [23], draw unbiased samples from the posterior distribution, but their sequential sampling method can be computationally expensive for both training and prediction. In addition, evaluating their convergence in finite time can be challenging [24].

While previous literature has explored various methods to approximate the nonmean-field Gaussian posterior, to the best of our knowledge, none has fully addressed the important problem of inducing point distribution in DGP inference. For instance, previous approaches, such as those proposed in [25], [26], and [27], attempted to design grids, orthogonal structures, or other special structures among inducing points. However, such structures may be handcrafted and may introduce bias by not fully capturing the information of the inducing points from the data. Other approaches, such as those based on normalizing flows [28], [29], face invertibility constraints that limit the flexibility of the transformation form of neural networks [30]. In addition, implicit distributions VI [31], [32], [33], [34],

Manuscript received 10 April 2023; revised 17 December 2023 and 29 February 2024; accepted 22 May 2024. Date of publication 15 August 2024; date of current version 7 April 2025. This work was supported in part by the Natural Science Foundation of Guangdong Province under Grant 2024A1515010089, Grant 2022A1515010179, and Grant 2023A1515011281; and in part by the National Natural Science Foundation of China under Grant 62272173 and Grant 61571005. *(Jian Xu and Shian Du contributed equally to this work.) (Corresponding authors: Qianli Ma; Delu Zeng.)*

Jian Xu is with the School of Mathematics, South China University of Technology, Guangzhou 510640, China (e-mail: 2713091379@qq.com).

Shian Du is with the Shenzhen International Graduate School, Tsinghua University, Shenzhen 100084, China (e-mail: dsa1458470007@gmail.com).

Junmei Yang and Delu Zeng are with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China (e-mail: yjunmei@scut.edu.cn; dlzeng@scut.edu.cn).

Qianli Ma is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: qianlima@scut.edu.cn).

Digital Object Identifier 10.1109/TNNLS.2024.3406635

[35], [36] attempts to estimate the difficult-to-handle non-Gaussian posterior variational lower bound through adversarial networks. However, controlling the bias and variance of the density ratio in high-dimensional space becomes exceedingly difficult, hindering the scalability and effectiveness of this approach [37], [38].

Therefore, in the context of high-dimensional nonmean-field scenarios, we propose a new VI framework for DGP based on Stein discrepancy (SD) [39], [40]. This is because SD provides accurate and efficient measures of distance between probability distributions, alleviating various computational issues in computing KL divergence. Unlike KL divergence, SD does not require computing the normalization constant of the distribution, and its gradient form often contains high-order information and geometric properties such as the curvature of the distribution, making it more effective for optimizing DGP high-dimensional nonmean-field posterior distributions and providing advantages in terms of error control and convergence rate [41], [42].

In this work, we introduce a novel inference framework for DGP models called neural operator VI (NOVI), which utilizes operators to optimize a regularized SD (RSD) with data subsampling. Specifically, we use Gaussian noise to transform a simple low-dimensional distribution into a high-dimensional complex distribution through a neural network generator, and then minimize the RSD between the generated distribution and the true posterior distribution using an SGD-based approach to obtain the gradients of the generator. The NOVI approach solves a minimax problem by Monte Carlo estimation and offers a black-box algorithmic solution that can handle complex posterior distributions for DGP models.

The main contributions are as follows.

1) We propose NOVI for DGPs, a novel variational framework based on SD and operator VI (OVI) with a neural generator. It minimizes RSD in $\mathcal{L}_2$ space between the generated distribution and true posterior to construct a more flexible and wider class of posterior approximations overcoming previous limitations caused by mean-field Gaussian posterior assumptions and the issues of nonmean-field minimization of KL divergence in the context of high-dimensional inducing points scenarios.
2) We provide theoretical evidence that our training schedule is essentially optimizing the Fisher divergence between the generated distribution and the true posterior distribution. In addition, the bias introduced by our method can be effectively controlled by multiplying the Fisher divergence with a constant. This feature of our approach enables us to achieve robust error control, ensuring the stability and precision of the algorithm.
3) We have conducted experimental demonstrations on eight UCI regression datasets and image classification datasets, which include MNIST, Fashion-MNIST, and CIFAR-10. The results demonstrate remarkable performance and faster convergence speeds than state-of-the-art (SOTA) methods, validating the effectiveness of the proposed model. By employing a convolutional architecture, we have achieved a classification accuracy of 93.56% on the CIFAR-10 dataset, surpassing the performance of SOTA GP methods.

Our code which encompasses comprehensive details and the full implementation of our NOVI-DGP approach can be accessed on our public GitHub repository https://github.com/studying910/NOVI-DGP.

II. Preliminary

In this section, we present necessary notations and settings on single-layer GPs and DGPs, and then we point out the flaws of the current model and introduce our motivation.

A. Gaussian Processes

Let a random function $f: \mathbb{R}^D \to \mathbb{R}$ map N training inputs $\mathbf{X} \triangleq \{\mathbf{x}_n\}_{n=1}^N$ to a collection of noisy observed outputs $\mathbf{y} \triangleq \{y_n\}_{n=1}^N$. In general, a zero mean GP prior is imposed on the function f, i.e., $f \sim \mathcal{GP}(0, k)$ where k represents a covariance 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 $\mathbf{X}$. This assumption yields a multivariate Gaussian prior over the function values $p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K_{XX}})$ where $[\mathbf{K_{XX}}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. In this work, we suppose $\mathbf{y}$ is contaminated by an i.i.d noise, thus $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I})$ where σ^2 is the noise variance. The GP posterior of the latent output $p(\mathbf{f}|\mathbf{y})$ has a closed-form solution [1] but suffers from $\mathcal{O}(N^3)$ computational cost and $\mathcal{O}(N^2)$ storage requirement, thus limiting its scalability to big data.

Advanced sparse methods have been developed to set so-called inducing points $\mathbf{z} = \{\mathbf{z}_m\}_{m=1}^M$ from the input space and the associated inducing outputs known as *inducing variables*: $\mathbf{u} = \{u_m = f(\mathbf{z}_m)\}_{m=1}^M$ [10], [11], [43], with time complexity of $\mathcal{O}(NM^2)$. In this SGPs paradigm [43], inducing variables $\mathbf{u}$ share a joint multivariate Gaussian distribution with $\mathbf{f}$: $p(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u})p(\mathbf{u})$ where the condition is specified as

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

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

To solve the intractable posterior distribution of inducing variables $p(\mathbf{u}|\mathbf{y})$, sparse variational GPs (SVGPs) [14], [43] reformulate the posterior inference problem as VI and confine the variational distribution to be $q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u})$ [16], [17], [22], [43]. This method approximates $q(\mathbf{u}) = \mathcal{N}(\mathbf{m}, \mathbf{S})$ [14], [15], [16], [17], [18], [19], [20], then a Gaussian marginal[1] is obtained by maximizing the evidence lower bound (ELBO) [44].

B. Deep GPs

A multilayer DGP model is a hierarchical composition of GP models constructed by stacking the multioutput SGPs together [45]. Consider a model with L layers and D_ℓ independent random functions in layer $\ell = 1, \ldots, L$ such that output of the $(\ell-1)$th layer $\mathbf{F}_{\ell-1}$ is used as an input to the ℓth layer, i.e., $\mathbf{F}_\ell \triangleq \{\mathbf{F}_{\ell,1} = f_{\ell,1}(\mathbf{F}_{\ell-1}), \ldots, \mathbf{F}_{\ell,D_\ell} = f_{\ell,D_\ell}(\mathbf{F}_{\ell-1})\}$, 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 corresponding inducing variables for each layer are denoted by $\mathcal{Z} \triangleq \{\mathbf{Z}_\ell\}_{\ell=1}^L$ and $\mathcal{U} \triangleq \{\mathbf{U}_\ell\}_{\ell=1}^L$, respectively,

[1]The solution is given in Appendix A.

where $\mathbf{U}_\ell \triangleq \{\mathbf{U}_{\ell,1} = f_{\ell,1}(\mathbf{Z}_\ell), \ldots, \mathbf{U}_{\ell,D_\ell} = f_{\ell,D_\ell}(\mathbf{Z}_\ell)\}$. Let $\mathcal{F} \triangleq \{\mathbf{F}_\ell\}_{\ell=1}^{L}$, the DGP model design yields the following joint model density:

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

Here, we place independent GP priors within and across layers on $\mathcal{U}$: $p(\mathcal{U}) = \prod_{l=1}^{L} p(\mathbf{U}_l) = \prod_{l=1}^{L} \prod_{d=1}^{D_\ell} \mathcal{N}(\mathbf{U}_{\ell,d}|0, \mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell})$ and the condition similar to (1) is defined as follows:

$$p(\mathbf{F}_\ell \mid \mathbf{F}_{\ell-1}, \mathbf{U}_\ell) = \prod_{d=1}^{D_\ell} \mathcal{N}\big(\mathbf{F}_{\ell,d} \mid \mathbf{K}_{\mathbf{F}_{\ell-1}\mathbf{Z}_\ell}\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}^{-1}\mathbf{U}_{\ell,d}\,, \mathbf{K}_{\mathbf{F}_{\ell-1}\mathbf{F}_{\ell-1}} - \mathbf{K}_{\mathbf{F}_{\ell-1}\mathbf{z}_\ell}\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}^{-1}\mathbf{K}_{\mathbf{Z}_\ell\mathbf{F}_{\ell-1}}\big). \tag{3}$$

As an extension of VI with DGPs, DSVI [22] approximates the posterior by requiring the distribution across the inducing outputs to be a posteriori Gaussian and independent amongst distinct GPs to obtain an analytical ELBO (known as the mean-field assumption [44], [46], $q(\mathbf{U}_{\ell,1:D_\ell}) = \mathcal{N}(\mathbf{m}_{\ell,1:D_\ell}, \mathbf{S}_{\ell,1:D_\ell})$, where $\mathbf{m}_{\ell,1:D_\ell}$ and $\mathbf{S}_{\ell,1:D_\ell}$ are variational parameters. By iteratively sampling the layer outputs and utilizing the reparameterization trick [47], DSVI enables scalability to big datasets.

The variational posterior distribution $q(\mathcal{U})$ in traditional approximation approaches for DGP models assumes that the distribution follows a mean-field Gaussian, which simplifies the analytical marginalization of the inducing outputs. However, this assumption is overly strict and may limit the effectiveness and expressiveness of the model. By Bayes' Rule, the true posterior distribution can be expressed in a more complex form that is not necessarily Gaussian

$$p(\mathcal{U}|\mathbf{y}) = \frac{p(\mathcal{U})p(\mathbf{y}|\mathcal{U})}{p(\mathbf{y})} = \frac{\int p(\mathbf{y}, \mathcal{F}, \mathcal{U}) d\mathcal{F}}{p(\mathbf{y})} \tag{4}$$

where $d\mathcal{F} = d\mathbf{F}_1 d\mathbf{F}_2, \ldots, d\mathbf{F}_L$. In DGP models, the likelihood term $p(\mathbf{y}|\mathcal{U})$ in the posterior (4) is difficult to compute because the latent functions $\mathbf{F}_1, \ldots, \mathbf{F}_{L-1}$ are inputs to a nonlinear kernel function. In addition, empirical evidence suggests that the true posterior distribution $p(\mathcal{U}|\mathbf{y})$ is often non-Gaussian, which makes it even more challenging to compute.

To address this issue, we introduce a novel variational family that balances computational efficiency and improved expressiveness, while also ensuring accurate error control, based on the concept of OVI [48]. Furthermore, our approach includes the learning of preservable transformations and the generation of approximate posterior samples through neural networks, as detailed in Sections III and IV.

III. OVI AND SD

Before using OVI and SD to develop a unique inference strategy for the DGP model, we provide a quick introduction to these concepts that form the foundation of our method.

Definition 1 [48]: Let $p(\boldsymbol{x})$ be a probability density supported on $\mathcal{X} \subseteq \mathbb{R}^d$ and $\boldsymbol{\phi}: \mathcal{X} \to \mathbb{R}^d$ be a differentiable function, we define Langevin–Stein operator (LSO) as

$$\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{x}) \triangleq (\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}))^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + \mathrm{Tr}(\nabla_{\boldsymbol{x}}\boldsymbol{\phi}(\boldsymbol{x})). \tag{5}$$

Lemma 1 [39]: Let $p(\boldsymbol{x})$ be a probability density function supported on $\mathcal{X} \subseteq \mathbb{R}^d$, and $\boldsymbol{\phi}: \mathcal{X} \to \mathbb{R}^d$ be a differentiable function. Suppose that $\int_{\partial\mathcal{X}} p(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x})d\boldsymbol{x} = \mathbf{0}$, where $\partial\mathcal{X}$ represents the boundary of $\mathcal{X}$. Under these conditions, Stein's identity can be expressed as

$$\mathbb{E}_{\boldsymbol{x}\sim p}\big[\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{x})\big] = 0. \tag{6}$$

When considering the expectation of $\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{x})$ under $\boldsymbol{x} \sim q$, where $q(\boldsymbol{x})$ is another probability density supported on $\mathcal{X} \subseteq \mathbb{R}^d$, the implication of Lemma 1 is that for arbitrary $\boldsymbol{\phi}$, the expectation will not be necessarily equal to zero. Instead, the magnitude of $\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{x})$ under $\boldsymbol{x} \sim q$ reflects the difference between probability distributions p and q. Thus, we can define a discrepancy measure, referred to as SD, to capture the difference between the target distribution and its approximation.

Definition 2 (Stein's Discrepancy [39]): Let $p(\boldsymbol{x})$, $q(\boldsymbol{x})$ be probability densities supported on $\mathcal{X} \subseteq \mathbb{R}^d$. SD is defined as the maximum violation of Stein's identity in a proper function set $\mathcal{G}$ for any differentiable function $\boldsymbol{\phi}: \mathcal{X} \to \mathbb{R}^d$, i.e.,

$$\mathcal{S}(q, p) \triangleq \sup_{\boldsymbol{\phi}\in\mathcal{G}} \mathbb{E}_{\boldsymbol{x}\sim q}[\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{x})]. \tag{7}$$

The selection of the function set $\mathcal{G}$ is crucial here, as it determines the discriminative power and computational feasibility of the SD. Traditionally, $\mathcal{G}$ consists of functions with bounded Lipschitz norms. However, this approach poses a challenging functional optimization problem that is computationally intractable or demands special considerations. Similar to prior approaches [49], we adopt the $\mathcal{L}_2$ space as the function space $\mathcal{F}$ in the SD (7) and represent $\boldsymbol{\phi}$ with a neural network $\boldsymbol{\phi}_{\boldsymbol{\eta}}$ as a discriminator to maximize

$$\mathrm{LSD}(q, p; \boldsymbol{\eta}) \triangleq \mathbb{E}_{\boldsymbol{x}\sim q}\big[(\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}))^{\mathrm{T}}\boldsymbol{\phi}_{\boldsymbol{\eta}}(\boldsymbol{x}) + \mathrm{Tr}\big(\nabla_{\boldsymbol{x}}\boldsymbol{\phi}_{\boldsymbol{\eta}}(\boldsymbol{x})\big)\big] \tag{8}$$

with respect to the parameters $\boldsymbol{\eta}$ of the neural network. This approach, known as the learned SD (LSD) [49], uses neural networks as discriminators to parameterize $\boldsymbol{\phi}$ in the SD (7). However, neural networks are not inherently square-integrable and do not vanish at infinity. In order to satisfy the conditions of Stein's identity [39], an $\mathcal{L}_2$ regularizer is applied to the LSD with a regularization strength $\lambda \in \mathbb{R}^+$, resulting in a RSD

$$\mathrm{RSD}(q, p; \boldsymbol{\eta}) \triangleq \mathbb{E}_{\boldsymbol{x}\sim q}\big[(\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}))^{\mathrm{T}}\boldsymbol{\phi}_{\boldsymbol{\eta}}(\boldsymbol{x}) + \mathrm{Tr}\big(\nabla_{\boldsymbol{x}}\boldsymbol{\phi}_{\boldsymbol{\eta}}(\boldsymbol{x})\big)\big] - \lambda\mathbb{E}_{\boldsymbol{x}\sim q}\big[\boldsymbol{\phi}_{\boldsymbol{\eta}}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\phi}_{\boldsymbol{\eta}}(\boldsymbol{x})\big]. \tag{9}$$

In Bayesian posterior inference, we aim to approximate the true posterior p using an approximate posterior $q_{\boldsymbol{\theta}}$, parameterized by variational parameters $\boldsymbol{\theta} \in \Theta$, where Θ is a set of possible parameterizations. Stein divergence, as defined in (7), is often used as the objective function for the OVI algorithm [48]. OVI is a black-box algorithm that leverages operators to optimize any operator-based objective, with the benefits of data subsampling and the capability to operate with a wider class of posterior approximations that do not require tractable densities. By combining the parameterizations of the set Θ and the discriminator $\boldsymbol{\phi}_{\boldsymbol{\eta}}$, OVI solves a minimax problem to find the optimal variational parameters $\boldsymbol{\theta}^\star$, i.e.,

$$\boldsymbol{\theta}^\star = \arg\inf_{\boldsymbol{\theta}\in\Theta}\left(\sup_{\boldsymbol{\eta}} \mathbb{E}_{\boldsymbol{x}\sim q_{\boldsymbol{\theta}}}\big[\mathcal{A}_p\boldsymbol{\phi}_{\boldsymbol{\eta}}(\boldsymbol{x})\big]\right). \tag{10}$$

IV. DGPS WITH NOVI

In this section, we present the algorithmic design for performing Bayesian inference on the posterior $p(\mathcal{U}|\mathcal{D})$ of

DGPs. We adopt the notation introduced in Section II-B, where $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$ denotes the training dataset, $\mathcal{U} \triangleq \{\mathbf{U}_\ell\}_{\ell=1}^L$ represents the inducing variables, and $\boldsymbol{\nu}$ denotes the hyperparameters of the DGP model, including the inducing point locations, kernel hyperparameters, and noise variance.

A. Neural Network as Generators

Consider a reference distribution $q_0(\boldsymbol{\epsilon})$ that generates noise $\boldsymbol{\epsilon} \in \mathbb{R}^{d_0}$. We represent the neural network that generates the posterior distribution, along with its parameters, as $g_{\boldsymbol{\theta}}$. If a noise vector is passed through this network, the resulting distribution of generated samples can be expressed as $\mathcal{U} = g_{\boldsymbol{\theta}}(\boldsymbol{\epsilon})$. In summary, our setup is shown as follows:

$$\boldsymbol{\epsilon} \sim q_0(\boldsymbol{\epsilon}), \quad g_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}) = \mathcal{U} \sim q_{\boldsymbol{\theta}}(\mathcal{U}). \tag{11}$$

Compared to traditional machine learning methods, such as grid-based or orthogonal designs, neural networks are recognized for their superior capability to model distributions, enabling them to learn implicit posterior distributions from data. As generators, neural networks can transform simple distributions such as Gaussian or uniform distributions, making them highly versatile and widely used in deep generative models [31], [38], [50], [51], [52], [53], [54], [55], [56]. As mentioned earlier, the high-dimensional and nonmean-field nature of the posterior distribution in deep generative models makes KL divergence unsuitable as a measure for the fit between the generative distribution $q_{\boldsymbol{\theta}}(\mathcal{U})$ and the true posterior $p(\mathcal{U}|\mathcal{D})$. Therefore, using OVI and RSD to construct a better objective is a reasonable approach.

B. Training Schedule

In Section III, we provided a review of OVI, a method that uses the LSO to enable a more flexible representation of the posterior geometry beyond the commonly used Gaussian distribution in vanilla VI. We extend this technique to the context of inducing points posterior inference for DGP models by iteratively training the discriminator and generator parameters to optimize the fit of the posterior to the data. Using the definition of RSD, we can construct an objective whose expectation value is zero if and only if the true posterior $p(\mathcal{U}|\mathcal{D})$ and the approximate distribution $q(\mathcal{U})$ are equivalent. During training, our goal is to minimize this objective

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\nu}) = \max_{\boldsymbol{\eta}} (\mathrm{RSD}\,(q_{\boldsymbol{\theta}}(\mathcal{U}), p(\mathcal{U}|\mathcal{D}, \boldsymbol{\nu}); \boldsymbol{\phi}_{\boldsymbol{\eta}})) \tag{12}$$

with respective to $\boldsymbol{\theta}$ and $\boldsymbol{\nu}$. Based on (9), we observe that (12) is a minmax problem. To solve it, we need to find the supremum on the right-hand side of the equation while jointly optimizing the inducing points posterior distribution and other model parameters $\boldsymbol{\nu}$ such as the GP kernel parameters. To achieve this, we separate the optimization of the discriminator $\boldsymbol{\phi}_{\boldsymbol{\eta}}$ and generator $g_{\boldsymbol{\theta}}$ to enable optimal estimation of these parameters. Since the other model parameters $\boldsymbol{\nu}$ are point estimates, we utilize Monte Carlo sampling and maximum likelihood estimation to optimize them after optimizing the discriminator and generator. We present the main idea of our algorithm and its pseudocode in Algorithm 1 and refer to as *NOVI* for DGP models.

In our implementation, we utilize the Monte Carlo method to estimate the objective (12) and RSD (9)

$$\begin{aligned}\widehat{\mathrm{RSD}}(q_{\boldsymbol{\theta}}, p; \boldsymbol{\phi}_{\boldsymbol{\eta}}) = \frac{1}{K}\sum_{k=1}^{K}\big(&\nabla_{\mathcal{U}} \log p(\mathcal{U} \mid \mathcal{D}, \boldsymbol{\nu})^{\mathrm{T}}\big|_{\mathcal{U}=\mathcal{U}^k}\boldsymbol{\phi}_{\boldsymbol{\eta}}(\mathcal{U}^k) \\ &+ \mathbb{E}_{\omega\sim\mathcal{N}(0,\mathrm{I})}\left(\omega^{\mathrm{T}}\nabla_{\mathcal{U}}\boldsymbol{\phi}_{\boldsymbol{\eta}}(\mathcal{U})\big|_{\mathcal{U}=\mathcal{U}^k}\omega\right)\big) \\ &- \lambda\frac{1}{K}\sum_{k=1}^{K}\left(\boldsymbol{\phi}_{\boldsymbol{\eta}}(\mathcal{U}^k)^{\mathrm{T}}\boldsymbol{\phi}_{\boldsymbol{\eta}}(\mathcal{U}^k)\right)\end{aligned} \tag{13}$$

where $\boldsymbol{\phi}_{\boldsymbol{\eta}^*}$ is the supremum of RSD estimate and the gradient with $\boldsymbol{\theta}$ and $\boldsymbol{\nu}$ is computed via automatic differentiation. To compute the expensive divergence of $\boldsymbol{\phi}_{\boldsymbol{\eta}}$ in (13), we use the Hutchinson estimator [57], which provides a stochastic estimate of the trace of a matrix and reduces the time complexity from $\mathcal{O}(D^2)$ to $\mathcal{O}(D)$, where D is the dimensionality of the matrix. In Theorem 1, we prove that the score function $\nabla_{\mathcal{U}} \log p(\mathcal{U}|\mathcal{D}, \boldsymbol{\nu})$ can be evaluated by Monte Carlo method, which demonstrates that RSD is a suitable objective for updating the parameters of the generator network.

Algorithm 1 NOVI for DGP Models

Input: training data $\mathcal{D} = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$, penalty parameter λ, n_c number of iterations for training the discriminator, learning rate α, β, γ, M batch size, sample number K
Initialize discriminator $\boldsymbol{\eta}$, generator $\boldsymbol{\theta}$, DGP hyperparameters $\boldsymbol{\nu}$
repeat
  for $j = 1$ **to** n_c **do**
    Sample a minibatch $\{\boldsymbol{x}_i, y_i\}_{i=1}^M \sim \mathcal{D}$
    Generate i.i.d. standard normal distribution noise $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_K$ from q_0
    Generate sample $g_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_1), \ldots, g_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_K)$ from the generator
    Compute empirical loss $\widehat{\mathrm{RSD}}(q_{\boldsymbol{\theta}}, p; \boldsymbol{\phi}_{\boldsymbol{\eta}})$
    $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta} - \alpha\nabla_{\eta}\widehat{\mathrm{RSD}}(q_{\boldsymbol{\theta}}, p; \boldsymbol{\phi}_{\boldsymbol{\eta}})$
  end for
  Compute empirical loss $\widehat{\mathrm{RSD}}(q_{\boldsymbol{\theta}}, p; \phi_{\eta^*})$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \beta\ \nabla_{\boldsymbol{\theta}}\widehat{\mathrm{RSD}}(q_{\boldsymbol{\theta}}, p; \phi_{\eta^*})$
  $\boldsymbol{\nu} \leftarrow \boldsymbol{\nu} - \gamma\ \frac{1}{K}\sum_{k=1}^K \nabla_{\boldsymbol{\nu}} \log p(\mathbf{y}, \mathcal{U}^k|\boldsymbol{\nu})$
until $\boldsymbol{\theta}, \boldsymbol{\nu}$ converge

Theorem 1: Assuming that $\mathcal{U} \in \Omega$, $\boldsymbol{\nu} \in \Upsilon$ where Ω and Υ are both compact spaces. We can obtain an asymptotically unbiased estimator for the score function $\nabla_{\mathcal{U}} \log p(\mathcal{U}|\mathcal{D}, \boldsymbol{\nu})$ in (13), which converges in probability to the true value (detailed proof can be seen in Appendix B)

$$\begin{aligned}\nabla_{\mathcal{U}} \log p(\mathcal{U}|\mathcal{D}, \boldsymbol{\nu}) \approx &-(\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_\ell, \ldots, \boldsymbol{\Delta}_L)^{\top} \\ &+ \nabla_{\mathcal{U}} \log \sum_{s=1}^{S} p\left(\mathbf{y}|\widehat{\mathbf{F}}_L^{(s)}\right)\end{aligned} \tag{14}$$

where $\boldsymbol{\Delta}_\ell = ((\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}{}^{-1}\mathbf{U}_{\ell,1})^{\top}, \ldots, (\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}{}^{-1}\mathbf{U}_{\ell,d})^{\top}, \ldots, (\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}{}^{-1}\mathbf{U}_{\ell,D_\ell})^{\top})$ and $\widehat{\mathbf{F}}_{\ell,d}^{(s)} \sim \mathcal{N}(\mathbf{K}_{\widehat{\mathbf{F}}_{\ell-1}\mathbf{Z}_\ell}\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}^{-1}\mathbf{U}_{\ell,d}, \mathbf{K}_{\widehat{\mathbf{F}}_{\ell-1}\widehat{\mathbf{F}}_{\ell-1}} - \mathbf{K}_{\widehat{\mathbf{F}}_{\ell-1}\mathbf{Z}_\ell}\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}^{-1}\mathbf{K}_{\mathbf{Z}_\ell\widehat{\mathbf{F}}_{\ell-1}})$ for $\ell = 1, \ldots, L$, S is the number of samples involved in estimation.

In the Monte Carlo estimation of the log-likelihood function, bias can arise due to the logarithmic transformation of the likelihood function, which is not explicitly defined in the DGP

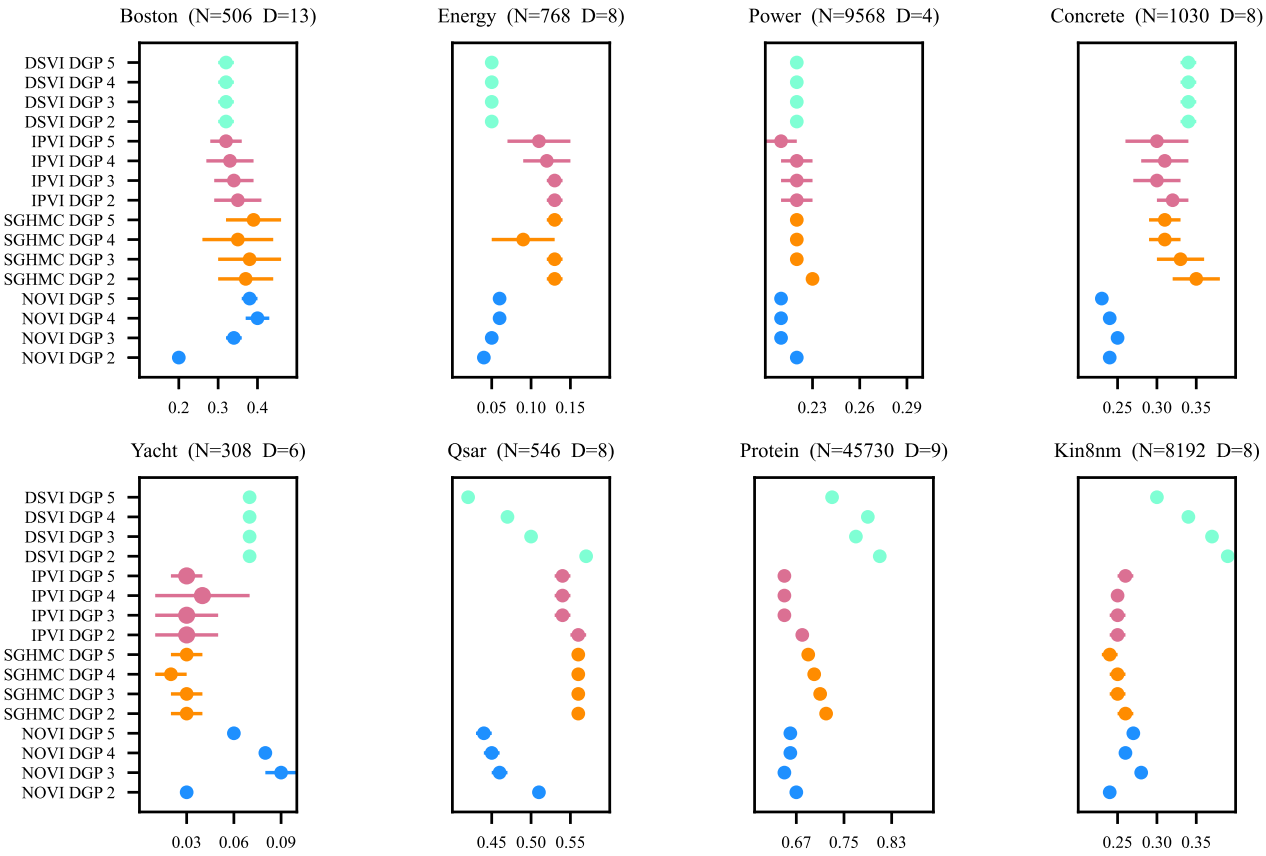

Fig. 1. Regression mean test RMSE results by our NOVI method (blue), SGHMC (orange), IPVI (pink), and DSVI (cyan) for DGPs on UCI benchmark datasets. The numbers 2, 3, 4, and 5 represent the layers of DGP methods. Lower is better. The mean is shown with error bars of one standard error.

model. This bias can affect other DGP inference objectives, such as DSVI [22].

However, in our method based on the score function, the gradient operator cancels out the bias introduced by the logarithmic transformation. The fact that our model is asymptotically unbiased, as proven by Theorem 1, is beneficial for mini-batch methods that rely on random subsampling of data. This property enhances both the scalability and accuracy of our method.

To ensure the convergence of this estimate, we propose to introduce a constraint function $c(\cdot)$ to restrict the parameter space of the objective function, which, to our knowledge, has not been considered in previous work. This constraint function $c(\cdot)$ is designed to confine $\mathcal{U}$ and $\boldsymbol{\nu}$ within a compact space. For instance, if we adopt a squared exponential kernel function $k_{\text{SE}}(x, x') = \sigma_f^2 \exp(-((x-x')^2)/2l^2)$ with length scale l, we can apply a clip function as a constraint for an appropriate closed interval $[P, Q]$, namely,

$$\text{clip}(l) = \begin{cases} P, & \text{if } l < P \\ l, & \text{if } P \leqslant l \leqslant Q \\ Q, & \text{if } l > Q. \end{cases} \tag{15}$$

As for the generated $\mathcal{U}$ by the neural network generator, we can also apply such a constraint in the last layer to ensure the convergence of the score function estimate.

C. Prediction

To obtain the final layer density for predicting the value of the test data $\mathcal{D}^\star = \{\mathbf{x}_n^\star, y_n^\star\}_{n=1}^{\text{T}}$, we first sample from the optimized generator and transform the input locations $\mathbf{x}_n$ to the test locations $\mathbf{x}_n^\star$ using (2). We subsequently compute the function values at the test locations, which are represented as $\mathbf{F}_\ell^\star$. Finally, we use (16) to estimate the density of the final layer, which enables us to make predictions for the test data

$$q(\mathbf{F}_L^\star) = \int \prod_{\ell=1}^{L} \prod_{d=1}^{D_\ell} p(\mathbf{F}_{\ell,d}^\star | \mathbf{F}_{\ell-1}^\star, \mathbf{U}_{\ell,d}) q_{\theta^\star}(\mathbf{U}_{\ell,d}) d\mathbf{F}_{\ell-1}^\star d\mathbf{U}_{\ell,d} \tag{16}$$

where $\boldsymbol{\theta}^\star$ is the optimal of the generator 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 consequence to draw samples from $q(\mathbf{F}_L^\star)$, and further perform the sampling using reparameterization trick [22], [58], [59]. Specifically, we first sample $\boldsymbol{\epsilon}^\ell \sim \mathcal{N}(0, \mathbf{I}_{D^\ell})$ and $\mathcal{U} \sim q_{\theta^\star}(\mathcal{U})$, then recursively draw the sampled variables $\widehat{\mathbf{F}}_{\ell,d}^\star \sim p(\mathbf{F}_{\ell,d}^\star | \widehat{\mathbf{F}}_{\ell-1}^\star, \mathbf{U}_{\ell,d})$ for $\ell = 1, \ldots, L$ as

$$\begin{aligned} \widehat{\mathbf{F}}_{\ell,d}^\star = {} & \mathbf{K}_{\widehat{\mathbf{F}}_{\ell-1}^\star \mathbf{Z}_\ell} \mathbf{K}_{\mathbf{Z}_\ell \mathbf{Z}_\ell}^{-1} \mathbf{U}_{\ell,d} \\ & + \boldsymbol{\epsilon}_\ell \odot \sqrt{\text{diag}\left(\mathbf{K}_{\widehat{\mathbf{F}}_{\ell-1}^\star \widehat{\mathbf{F}}_{\ell-1}^\star} - \mathbf{K}_{\widehat{\mathbf{F}}_{\ell-1}^\star \mathbf{Z}_\ell} \mathbf{K}_{\mathbf{Z}_\ell \mathbf{Z}_\ell}^{-1} \mathbf{K}_{\mathbf{Z}_\ell \widehat{\mathbf{F}}_{\ell-1}^\star}\right)} \end{aligned} \tag{17}$$

where "$\odot$" represents the Hadamard product, and the square root is element-wise, and we define $\mathbf{F}_0^\star \triangleq \mathbf{X}^\star$ for the first layer and use $\text{diag}(\cdot)$ to denote the vector of diagonal elements of a matrix. The diagonal approximation in (17) holds since in DGP model, the ith marginal of approximate posterior $q(\mathbf{F}_{(\ell,d)}^i)$ depends only on the corresponding inputs $\boldsymbol{x}_i$ [11].

TABLE I

REGRESSION TEST RMSE RESULTS FOR LARGE DATASETS

Data	DSVI 2	DSVI 3	DSVI 4	DSVI 5	SGHMC 2	SGHMC 3	SGHMC 4	SGHMC 5	IPVI 2	IPVI 3	IPVI 4	IPVI 5	NOVI 2	NOVI 3	NOVI 4	NOVI 5
Year	9.58	8.98	8.93	8.87	9.05	8.91	8.85	8.81	8.95	8.84	8.80	8.79	**8.84**	**8.79**	**8.76**	**8.74**
Airline	24.6	24.3	24.2	24.1	24.1	23.8	23.7	23.6	24.0	23.7	23.6	23.6	**23.8**	**23.6**	**23.5**	**23.5**

TABLE II

MEAN TEST ACCURACY (%) AND TRAINING DETAILS ACHIEVED BY DSVI, SGHMC, AND NOVI (OURS) DGP MODEL FOR THREE IMAGE CLASSIFICATION DATASETS. BATCH SIZE IS SET TO 256 FOR ALL METHODS. L DENOTES THE NUMBER OF HIDDEN LAYERS. OUR PROPOSED METHOD CAN ALSO BE COMBINED WITH CONVOLUTION KERNELS [69] TO OBTAIN A BETTER RESULT, FOR A FAIR COMPARISON, WE HAVE NOT IMPLEMENTED IT HERE BUT IN THE NEXT PART

Data Set	Model	Time (L=3)	Iter(L=3)	Acc (L=3)	Time (L=4)	Iter (L=4)	Acc (L=4)
MNIST	DSVI	0.34s/iter	20K	97.17	0.54s/iter	20K	97.41
	IPVI	0.49s/iter	20K	97.58	0.62s/iter	20K	97.80
	SGHMC	1.14s/iter	20K	97.25	1.22s/iter	20K	97.55
	NOVI (ours)	0.38s/iter	10K	**98.04**	0.50s/iter	10K	**98.21**
Fashion-MNIST	DSVI	0.34s/iter	20K	87.45	0.50s/iter	20K	87.99
	IPVI	0.48s/iter	20K	88.23	0.61s/iter	20K	88.90
	SGHMC	1.21s/iter	20K	86.88	1.25s/iter	20K	87.08
	NOVI (ours)	0.40s/iter	10K	**89.36**	0.55s/iter	10K	**89.65**
CIFAR-10	DSVI	0.43s/iter	20K	51.33	0.66s/iter	20K	51.79
	IPVI	0.62s/iter	20K	52.79	0.78s/iter	20K	53.27
	SGHMC	8.04s/iter	20K	52.34	8.61s/iter	20K	52.81
	NOVI (ours)	0.43s/iter	10K	**53.42**	0.52s/iter	10K	**53.62**

V. CONVERGENCE GUARANTEES AND BIAS CONTROL

In this section, we provide convergence guarantees and error control for NOVI, detailed proof can be seen in Appendix C.

Definition 3: The Fisher divergence [60] between two suitably smooth density functions is defined as

$$FD(q, p) = \int_{\mathbb{R}^d} \|\nabla \log q(\boldsymbol{x}) - \nabla \log p(\boldsymbol{x})\|_2^2 q(\boldsymbol{x}) d\boldsymbol{x}. \quad (18)$$

Theorem 2: Training the generator with the optimal discriminator corresponds to minimizing the Fisher divergence between $q_{\boldsymbol{\theta}}$ and p, and the corresponding optimal loss for (12) is

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\nu}) = \frac{1}{4\lambda} FD(q_{\boldsymbol{\theta}}(\mathcal{U}), p(\mathcal{U}|\mathcal{D}, \boldsymbol{\nu})) \quad (19)$$

where $\lambda \in \mathbb{R}^+$ is a regularization strength defined in (9).

Theorem 3: Assuming that $\mathcal{U} \in \Omega$, $\boldsymbol{\nu} \in \Upsilon$ where Ω and Υ are both compact spaces. The bias of the estimation for prediction $\widehat{\mathbf{F}}_L^{\star}$ in (17) from the DGPs exact evaluation can be bounded by the square root of the Fisher divergence between $q_{\boldsymbol{\theta}}(\mathcal{U})$ and $p(\mathcal{U}|\mathcal{D}, \boldsymbol{\nu})$ up to multiplying a constant.

Theorem 2 demonstrates that our algorithm is equivalent to minimizing Fisher divergence, while Theorem 3 guarantees a bounded bias for prediction estimation. Fisher divergence has proven to be a valuable tool in various statistics and machine learning applications, as demonstrated by its use in practical contexts such as generative models [61], Bayesian inference [60], and others [42].

Moreover, Fisher divergence has strong connections to other distance metrics such as total variation [62], Hellinger distance [63], and Wasserstein distance [64]. In fact, it is often a stronger distance metric than these alternatives [65]. This means that when the Fisher divergence is smaller, it implies that the other distance metrics, which are weaker than the Fisher divergence, will also be smaller. Consequently, using Fisher divergence as a measure provides better error control in comparison to these alternative metrics. The stronger connection of Fisher divergence to other distance metrics allows the proposed approach to capture more subtle differences between probability distributions, leading to more accurate moment estimates and improved performance, especially in high-dimensional distribution estimation [42]. By leveraging the advantages of Fisher divergence, our method achieves enhanced theoretical guarantees and translates them into practical gains in real-world tasks.

TABLE III

CONVOLUTIONAL RESULTS OF CIFAR10 DATASET COMPARED WITH BASELINE DEEP LEARNING AND DGP METHODS. OUR RESULTS INDICATE THAT OUR MODEL OUTPERFORMS RESNET WHEN COMPARED, WITH ONLY AN ADDITION OF LESS THAN ONE-TENTH OF THE PARAMETER COUNT

Models	Accuracy (%)
CNF [29]	76.8
BDCGP [75]	74.6
DCGP [76]	75.9
DKL[74]	77.0
Resnet-20	91.3
Resnet-56	93.03
NOVI-DGP	**93.56**

VI. RELATED WORKS

Our method for inference is inspired by two previous works, namely, OVI [48] and SD [39]. However, our approach is distinct in that it specifically focuses on the DGP posterior and develops tailored algorithms to address it. One key challenge in calculating the score function for DGPs is that the likelihood function is not explicit, and thus we propose a stochastic gradient and Monte Carlo sampling method to address this issue (see Theorem 1). While OVI [48] introduces a similar objective for inference to RSD, it utilizes a different class of discriminator and does not employ many of the SOTA techniques we use for scalability, such as the Hutchinson estimator [57].

The computation of the term $\text{Tr}(\nabla_{\boldsymbol{x}} \boldsymbol{\phi}_{\boldsymbol{\eta}}(\boldsymbol{x}))$ in (8) is computationally expensive as it requires $\mathcal{O}(d)$ vector-Jacobian products since each entry of the diagonal of the Jacobian requires

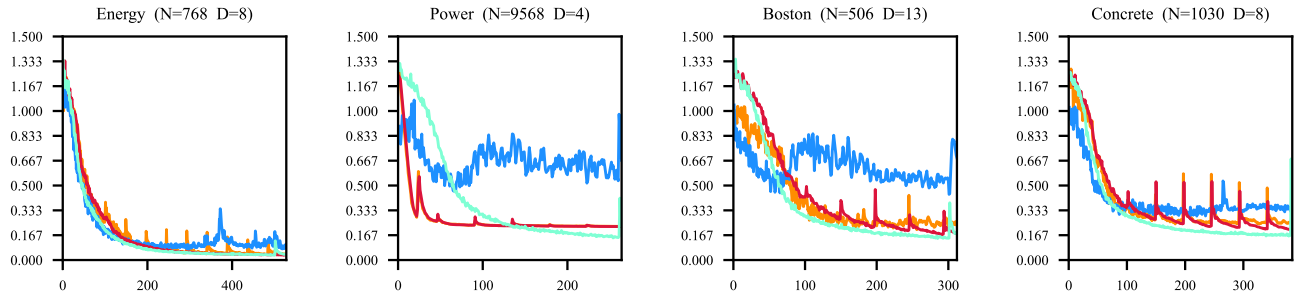

Fig. 2. Mean RMSE comparison of NOVI (test: orange, train: red) with Monte Carlo log-likelihood maximization method (test: blue, train: cyan) using two-layer DGP model on four UCI regression datasets.

computing a separate derivative of $\boldsymbol{\phi}_\eta$. To address this issue, we can use the Hutchinson estimator, which only requires one vector-Jacobian product to compute. This estimator can be obtained by multiplying the matrix $\nabla_{\boldsymbol{x}}\boldsymbol{\phi}_\eta(\boldsymbol{x})$ by a noise vector twice, as shown in the following identity [57]:

$$\mathrm{Tr}\big(\nabla_{\boldsymbol{x}}\boldsymbol{\phi}_\eta(\boldsymbol{x})\big) = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,I)}\big[\boldsymbol{\epsilon}^{\mathrm{T}}\nabla_{\boldsymbol{x}}\boldsymbol{\phi}_\eta(\boldsymbol{x})\boldsymbol{\epsilon}\big]. \tag{20}$$

This single-sample Monte-Carlo estimator has been widely used in recent years in the machine learning community due to its efficiency and unbiasedness [66], [67], [68]. The basic principle of this estimator is to approximate the trace by introducing random variables, such as Gaussian distributed vectors. Specifically, we sample the matrix using multiple random vectors and estimate the trace of the matrix-vector product by summing the products of the sampled results. The benefit of this estimation is that it only requires computing the products of the matrix-vector multiplications, rather than explicitly computing the trace of the entire matrix, thereby reducing computational complexity.

TABLE IV

COMPARISON OF TRAINING TIME (S) OF A SINGLE ITERATION AND TOTAL TRAINING ITERATIONS ON ENERGY DATASET. BATCH SIZE IS SET TO 1000 FOR ALL FOUR METHODS

Type	DSVI 2	DSVI 3	DSVI 4	DSVI 5
Time (s)	0.835	0.903	0.965	1.339
Iteration	20K	20K	20K	20K
Type	IPVI 2	IPVI 3	IPVI 4	IPVI 5
Time (s)	0.117	0.162	0.211	0.260
Iteration	20K	20K	20K	20K
Type	SGHMC 2	SGHMC 3	SGHMC 4	SGHMC 5
Time (s)	0.630	1.000	1.490	1.870
Iteration	20K	20K	20K	20K
Type	NOVI 2	NOVI 3	NOVI 4	NOVI 5
Time (s)	0.391	0.613	0.863	1.123
Iteration	500	500	500	500

VII. EXPERIMENTS

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. Our method was compared against several other models, including DSVI [22], which was used as our baseline model, implicit posterior VI (IPVI) [33], which constructs a variational lower bound using density ratio estimates, and the SOTA SGHMC model [23]. All experiments were conducted with the same hyper-parameters and initializations, and we provide detailed training information in Appendix E.

A. UCI Regression Benchmark

In our experiments, we evaluated the performance of the NOVI model on eight UCI regression datasets, which varied in size from 308 to 45 730 data points. We used the average RMSE of the test data as the performance metric, and the results are presented in Fig. 1. The tabular version of the results can be found in Appendix D-C.

As shown in Fig. 1, our NOVI method consistently achieves competitive results compared to baselines on the majority of datasets. This is attributed to the key advantages of our approach, which overcomes limitations present in previous methods such as the restrictive nature of mean-field posterior distributions, and provides stronger guarantees in terms of convergence and error control using Fisher divergence, as discussed in Section V.

In the analysis of the "Boston," "Energy," "Yacht," and "Kin8nm" datasets, our method demonstrates impressive performance with two-layer DGPs. As the number of layers increases, there is some uncertainty observed. However, when compared to other baseline methods, NOVI remains competitive. Notably, for larger datasets like "Power" and "Protein," deeper NOVI models exhibit superior performance compared to other methods. These results suggest that the effectiveness of our method may vary depending on the dataset characteristics and the optimal model hyperparameters. It is essential to note that performance fluctuations with increasing layers are not exclusive to our model but are also evident in other models such as SGHMC and IPVI.

We have also included additional results for real-world regression datasets in Appendix D-E, further demonstrating the effectiveness of the NOVI model.

B. Large-Scale Regression

Using a mini-batch algorithm, our method can also be extended to datasets at the million level. Our evaluation of the performance of NOVI is 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 the next 100K for testing.

In these regression tasks, the performance metric used is the RMSE of the test data. Table I presents the results of the test RMSE and the standard deviation over 10 runs. Notably, it can be observed that NOVI generally outperforms SGHMC, DSVI, and IPVI. Particularly for large-scale regression tasks, the performance of NOVI consistently improves with increasing depth. This observation signifies that our NOVI model maintains its superiority even for very large datasets with millions of data points.

C. Image Classification

We evaluate our method on multiclass classification tasks using the MNIST [70], Fashion-MNIST [71], and CIFAR-10 [72] 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 II. We note that our method outperforms the other three methods on all three datasets, with significantly less training time and iterations. In addition, we evaluate our approach using three UCI classification datasets, and the results are presented in Appendix D-A.

Furthermore, we conduct supplementary experiments to achieve superior performance on the CIFAR-10 dataset. We utilize the convolutional layers of ResNet-20 [73] as our feature extractor and achieve a remarkable accuracy of 93.56 on the test set under pretraining [74], surpassing the performance of all baseline methods, as detailed in Table III.

D. Computational Complexity

We compared the training efficiency of our model with three other methods on a single Tesla V100 GPU using the Energy dataset, and the results are presented in Table IV. It can be observed that our model achieves faster iteration times compared to DSVI and SGHMC and requires less time among the other methods. Furthermore, our method achieves convergence in less than one-tenth of the number of iterations required by the other three methods. In addition, Table II shows that NOVI requires significantly less training time and iterations to converge on high-dimensional image datasets, demonstrating the scalability of our proposed method to larger datasets. Details regarding the number of inducing points used in each method can be found in Appendix D-D.

E. Ablation Study

To demonstrate the effectiveness of our proposed NOVI method, we compare it with a two-layer DGP model by directly maximizing the log-likelihood with randomly initialized $\mathcal{U}$ and hyperparameters $\boldsymbol{\nu}$. The results are presented in Fig. 2. It can be observed that NOVI achieves lower test RMSE and higher train RMSE for all datasets, which indicates that our optimization method reduces overfitting. Although there is some loss fluctuation during the training of our method, it is caused by the unique adversarial training and converges to a stable value after only several hundred iterations. Additional results for the ablation study can be found in the Appendix.

VIII. Conclusion

This article introduces a novel framework called NOVI, which integrates the SD with DGPs to model non-Gaussian and hierarchical-related posteriors, thereby enhancing the flexibility of DGP models. The approach involves generating inducing variables from a neural generator and optimizing them jointly with variational parameters through adversarial training. Theoretical analysis shows that the bias introduced by our method can be bounded by Fisher divergence, enabling efficient optimization of the neural generator.

Empirical evaluation indicates that NOVI outperforms SOTA approximation methods for both regression and classification tasks while requiring significantly less training time and iterations to converge. We validated our model on 18 publicly available datasets, including 6 classification datasets and 12 regression datasets. These datasets range in sample size from hundreds (e.g., Boston) to millions (e.g., Year), and are mostly from real-world scenarios. Our model outperformed the latest five baseline methods on 16 of these datasets. The improvements, particularly in metrics like mse, are considerable, indicating the effectiveness of our model from a hypothesis-testing perspective. We also observed performance fluctuations in some datasets as the number of DGP layers increased during experiments. We hope that future work can address this limitation by making technical breakthroughs in this area.

Future work could also focus on utilizing neural architecture search (NAS) [77] methods to obtain more suitable network architectures for practical applications. Overall, the proposed NOVI framework represents a significant advancement in the field of deep learning and holds promise for a wide range of applications in both academia and industry.

Appendix A
Solution to SVGP and DSVI

Due to the Gaussian mean-field assumptions, the solution to SVGP has an analytical solution

$$\begin{aligned} q(\mathbf{f}) &= \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \text{where} \\ \boldsymbol{\mu} &= \mathbf{K}_{\mathbf{XZ}}\mathbf{K}_{\mathbf{ZZ}}^{-1}\mathbf{m} \\ \boldsymbol{\Sigma} &= \mathbf{K}_{\mathbf{XX}} - \mathbf{K}_{\mathbf{XZ}}\mathbf{K}_{\mathbf{ZZ}}^{-1}(\mathbf{K}_{\mathbf{ZZ}} - \mathbf{S})\mathbf{K}_{\mathbf{ZZ}}^{-1}\mathbf{K}_{\mathbf{ZX}}. \end{aligned} \tag{21}$$

While performing similarly in DSVI, they have an analytical form for $q(\mathbf{F})$

$$\begin{aligned} q(\{\mathbf{F}_\ell\}_{\ell=1}^L) &= \prod_{\ell=1}^{L}\prod_{d=1}^{D_\ell}\int q\left(\mathbf{F}_{\ell,d}|\mathbf{F}_{\ell-1}, \mathbf{U}_{\ell,d}\right)q\left(\mathbf{U}_{\ell,d}\right)d\mathbf{U}_{\ell,d} \\ &= \prod_{\ell=1}^{L}\prod_{d=1}^{D_\ell}\mathcal{N}\left(\mathbf{F}_{\ell,d}|\boldsymbol{\mu}_{\ell,d}, \boldsymbol{\Sigma}_{\ell,d}\right) \end{aligned} \tag{22}$$

where $\boldsymbol{\mu}_{\ell,d}, \boldsymbol{\Sigma}_{\ell,d}$ is defined as (21).

Appendix B
Proof of Theorem 1

Theorem 4: Assuming that $\mathcal{U} \in \Omega$, $\nu \in \Upsilon$ where Ω and Υ are both compact spaces. We can obtain an asymptotically unbiased estimator for the score function $\nabla_{\mathcal{U}} \log p(\mathcal{U}|\mathcal{D}, \nu)$

in (18), which converges in probability to the true value

$$\nabla_{\mathcal{U}} \log p(\mathcal{U}|\mathcal{D}, \nu) \approx -(\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_\ell, \ldots, \boldsymbol{\Delta}_L)^\top + \nabla_{\mathcal{U}} \log \sum_{s=1}^{S} p\left(\mathbf{y}|\widehat{\mathbf{F}}_L^{(s)}\right) \tag{23}$$

where $\boldsymbol{\Delta}_\ell = ((\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}{}^{-1}\mathbf{U}_{\ell,1})^\top, \ldots, (\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}{}^{-1}\mathbf{U}_{\ell,d})^\top, \ldots, (\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}{}^{-1}\mathbf{U}_{\ell,D_\ell})^\top)$ and $\widehat{\mathbf{F}}_{\ell,d}^{(s)} \sim \mathcal{N}(\mathbf{K}_{\widehat{\mathbf{F}}_{\ell-1}\mathbf{Z}_\ell}\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}^{-1}\mathbf{U}_{\ell,d}, \mathbf{K}_{\widehat{\mathbf{F}}_{\ell-1}\widehat{\mathbf{F}}_{\ell-1}} - \mathbf{K}_{\widehat{\mathbf{F}}_{\ell-1}\mathbf{Z}_\ell}\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}^{-1}\mathbf{K}_{\mathbf{Z}_\ell\widehat{\mathbf{F}}_{\ell-1}})$ for $\ell = 1, \ldots, L$, S is the number of samples involved in estimation.

Proof: From Bayes formula

$$\log p(\mathcal{U}|\mathcal{D}, \nu) = \log \frac{p(\mathcal{U})p(\mathcal{D}|\mathcal{U}, \nu)}{p(\mathcal{D})} = \log p(\mathcal{U}) + \log p(\mathcal{D}|\mathcal{U}, \nu) - \log p(\mathcal{D}) \tag{24}$$

since the prior term $p(\mathbf{U}_{\ell,d}) = \mathcal{N}(0, \mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell})$, the gradient with $\mathcal{U}$ is a long vector and is tractable

$$\begin{aligned}\nabla_{\mathcal{U}} \log p(\mathcal{U}) &= \nabla_{\mathcal{U}} \log \prod_{\ell=1}^{L}\prod_{d=1}^{D_\ell} p\left(\mathbf{U}_{\ell,d}\right) \\ &= -\frac{1}{2}\sum_{\ell=1}^{L}\sum_{d=1}^{D_\ell} \nabla_{\mathcal{U}} \mathbf{U}_{\ell,d}^T \mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}^{-1}\mathbf{U}_{\ell,d} \\ &= -(\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_\ell, \ldots, \boldsymbol{\Delta}_L)^\top. \end{aligned} \tag{25}$$

The third term of (24) is a constant with respect to $\mathcal{U}$. We compute the second data likelihood term $\log p(\mathcal{D}|\mathcal{U}, \nu)$ using the reparameterization trick and Monte Carlo method over each layer

$$\begin{aligned}&\nabla_{\mathcal{U}} \log p(\mathcal{D}|\mathcal{U}, \nu) \\ &= \nabla_{\mathcal{U}} \log \int p(\mathbf{y}|\mathbf{F}_L) \prod_{\ell=1}^{L} p(\mathbf{F}_\ell|\mathbf{F}_{\ell-1}, \mathbf{U}_\ell) d\mathbf{F}_{\ell-1} \\ &= \nabla_{\mathcal{U}} \log \mathbb{E}_{p(\mathbf{F}_L|\mathcal{U})}\ p(\mathbf{y}|\mathbf{F}_L) \approx \nabla_{\mathcal{U}} \log \sum_{s=1}^{S} p\left(\mathbf{y}|\widehat{\mathbf{F}}_L^{(s)}\right). \end{aligned} \tag{26}$$

The last equation in the above expression can be derived from the following conditions: we denote Monte Carlo estimation $(1/S)\sum_{s=1}^{S} p(\mathbf{y}|\widehat{\mathbf{F}}_L^{(s)})$ as $\tilde{p}(\mathbf{y}|\mathcal{U})$ and the true value as $p(\mathbf{y}|\mathcal{U})$, respectively. By the Central Limit Theorem, $(\tilde{p}(\mathbf{y}|\mathcal{U}) - p(\mathbf{y}|\mathcal{U}))/(((1/S)\mathrm{Var}(p(\mathbf{y}|\widehat{\mathbf{F}}_L^{(s)})))^{1/2}) \sim \mathcal{N}(0, 1)$, i.e., $\tilde{p}(\mathbf{y}|\mathcal{U}) \xrightarrow{\mathcal{P}} p(\mathbf{y}|\mathcal{U})$ and $\nabla\tilde{p}(\mathbf{y}|\mathcal{U}) \xrightarrow{\mathcal{P}} \nabla p(\mathbf{y}|\mathcal{U})$, since $\nabla\tilde{p}(\mathbf{y}|\mathcal{U}) - \nabla p(\mathbf{y}|\mathcal{U}) = \nabla((\mathrm{Var}(p(\mathbf{y}|\widehat{\mathbf{F}}_L^{(s)})))/S)^{1/2}\epsilon \xrightarrow{\mathcal{P}} 0$ as S increases, where $\epsilon \sim \mathcal{N}(0, 1)$. The likelihood function $p(\mathbf{y}|U, \nu)$ is a continuous bounded function defined on a compact domain Ω and Υ, then uniform continuity guarantees the boundedness of its derivative, then we have (27), as shown at the bottom of the next page.

It is easy to derive from the above equation that $\nabla \log \tilde{p}(\mathbf{y} \mid \mathcal{U}) \xrightarrow{\mathcal{P}} \nabla \log p(\mathbf{y}|\mathcal{U})$, the approximately equal sign means that the right-hand side converges to the left-hand side in probability. From the above expression, we can also conclude that this estimator is asymptotically unbiased.

We draw S samples $\widehat{\mathbf{F}}_{\ell,d}^{(s)}$ from $\widehat{\mathbf{F}}_{\ell,d} \sim p(\mathbf{F}_{\ell,d}|\widehat{\mathbf{F}}_{\ell-1}, \mathbf{U}_{\ell,d})$ for $\ell = 1, \ldots, L$ as

$$\widehat{\mathbf{F}}_{\ell,d} = \mathbf{K}_{\widehat{\mathbf{F}}_{\ell-1}\mathbf{Z}_\ell}\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}^{-1}\mathbf{U}_{\ell,d} + \epsilon_\ell \odot \sqrt{\mathrm{diag}\left(\mathbf{K}_{\widehat{\mathbf{F}}_{\ell-1}\widehat{\mathbf{F}}_{\ell-1}} - \mathbf{K}_{\widehat{\mathbf{F}}_{\ell-1}\mathbf{Z}_\ell}\mathbf{K}_{\mathbf{Z}_\ell\mathbf{Z}_\ell}^{-1}\mathbf{K}_{\mathbf{Z}_\ell\widehat{\mathbf{F}}_{\ell-1}}\right)} \tag{28}$$

where $\epsilon^\ell \sim \mathcal{N}(0, \mathbf{I}_{D^\ell})$. As a result, we obtain the score function via automatic differentiation

$$\nabla_{\mathcal{U}} \log p(\mathcal{U}|\mathcal{D}, \boldsymbol{\nu}) \approx -(\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_\ell, \ldots, \boldsymbol{\Delta}_L)^\top + \nabla_{\mathcal{U}} \log \sum_{s=1}^{S} p\left(\mathbf{y}|\widehat{\mathbf{F}}_L^{(s)}\right). \tag{29}$$

APPENDIX C
PROOF OF THEOREMS 2 AND 3

Definition 4: Let $p(\boldsymbol{x})$ be a probability density supported on $\mathcal{X} \subseteq \mathbb{R}^d$ and $\boldsymbol{\phi}: \mathcal{X} \to \mathbb{R}^d$ be a differentiable function, we define LSO [48]

$$\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + \mathrm{Tr}(\nabla_{\boldsymbol{x}}\boldsymbol{\phi}(\boldsymbol{x})). \tag{30}$$

Lemma 2: Let $p(\boldsymbol{x})$ be a probability density function supported on $\mathcal{X} \subseteq \mathbb{R}^d$, and $\boldsymbol{\phi}: \mathcal{X} \to \mathbb{R}^d$ be a differentiable function. Suppose that $\int_{\partial\mathcal{X}} p(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x})d\boldsymbol{x} = \mathbf{0}$, where $\partial\mathcal{X}$ represents the boundary of $\mathcal{X}$. Under these conditions, Stein's identity can be expressed as

$$\mathbb{E}_{\boldsymbol{x}\sim p}\left[\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{x})\right] = 0. \tag{31}$$

Proof:

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}\sim p}[\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{x})] \\ &= \mathbb{E}_{\boldsymbol{x}\sim p}[\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + \mathrm{Tr}(\nabla_{\boldsymbol{x}}\boldsymbol{\phi}(\boldsymbol{x}))] \\ &= \mathrm{Tr}(\mathbb{E}_{\boldsymbol{x}\sim p}[\boldsymbol{\phi}(\boldsymbol{x})\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})^{\mathrm{T}} + \nabla_{\boldsymbol{x}}\boldsymbol{\phi}(\boldsymbol{x})]) \end{aligned} \tag{32}$$

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}\sim p}[\boldsymbol{\phi}(\boldsymbol{x})\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})^{\mathrm{T}} + \nabla_{\boldsymbol{x}}\boldsymbol{\phi}(\boldsymbol{x})] \\ &= \int_{\mathcal{X}} p(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x})\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})^{\mathrm{T}} + p(\boldsymbol{x})\nabla_{\boldsymbol{x}}\boldsymbol{\phi}(\boldsymbol{x})d\boldsymbol{x} \\ &= \int_{\mathcal{X}} \nabla_{\boldsymbol{x}}(p(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x}))d\boldsymbol{x}. \end{aligned} \tag{33}$$

From divergence theorem

$$\begin{aligned}&\mathrm{Tr}\left(\int_{\mathcal{X}} \nabla_{\boldsymbol{x}}(p(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x}))d\boldsymbol{x}\right) \\ &= \int_{\mathcal{X}} \mathrm{div}(p(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x}))d\boldsymbol{x} \\ &= \int_{\partial\mathcal{X}} p(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{n}(\boldsymbol{x})d\boldsymbol{x} = 0 \end{aligned} \tag{34}$$

where $\boldsymbol{n}(\boldsymbol{x})$ is the outward-pointing unit vector on the boundary of $\mathcal{X}$.

Lemma 3: Suppose $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ are probability densities supported on $\mathcal{X} \subseteq \mathbb{R}^d$ and $\boldsymbol{\phi}: \mathcal{X} \to \mathbb{R}^d$ is a differentiable function satisfying $\int_{\partial\mathcal{X}} p(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x})d\boldsymbol{x} = \mathbf{0}$ and $\int_{\partial\mathcal{X}} q(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x})d\boldsymbol{x} = \mathbf{0}$, then

$$\mathbb{E}_{\boldsymbol{x}\sim q}\left[\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{x})\right] = \mathbb{E}_{\boldsymbol{x}\sim q}\left[(\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x}))^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})\right]. \tag{35}$$

Proof: By Lemma 1

$$\mathbb{E}_{\boldsymbol{x}\sim q}\left[\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + \mathrm{Tr}(\nabla_{\boldsymbol{x}}\boldsymbol{\phi}(\boldsymbol{x}))\right] = 0$$
$$\Rightarrow \mathbb{E}_{\boldsymbol{x}\sim q}[\mathrm{Tr}(\nabla_{\boldsymbol{x}}\boldsymbol{\phi}(\boldsymbol{x}))] = -\mathbb{E}_{\boldsymbol{x}\sim q}\left[\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})\right] \tag{36}$$

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}\sim q}\left[\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{x})\right] \\ &= \mathbb{E}_{\boldsymbol{x}\sim q}[\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + \mathrm{Tr}(\nabla_{\boldsymbol{x}}\boldsymbol{\phi}(\boldsymbol{x}))] \\ &= \mathbb{E}_{\boldsymbol{x}\sim q}[(\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x}))^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})]. \end{aligned} \tag{37}$$

Lemma 4: For any $\boldsymbol{a}, \boldsymbol{y} \in \mathbb{R}^d$ and $\lambda > 0$, the function $\boldsymbol{y} \mapsto \boldsymbol{a}^{\mathrm{T}}\boldsymbol{y} - \lambda\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}$ achieves its maximum $(1/(4\lambda))\boldsymbol{a}^{\mathrm{T}}\boldsymbol{a}$ if and only if $\boldsymbol{y} = (1/(2\lambda))\boldsymbol{a}$.

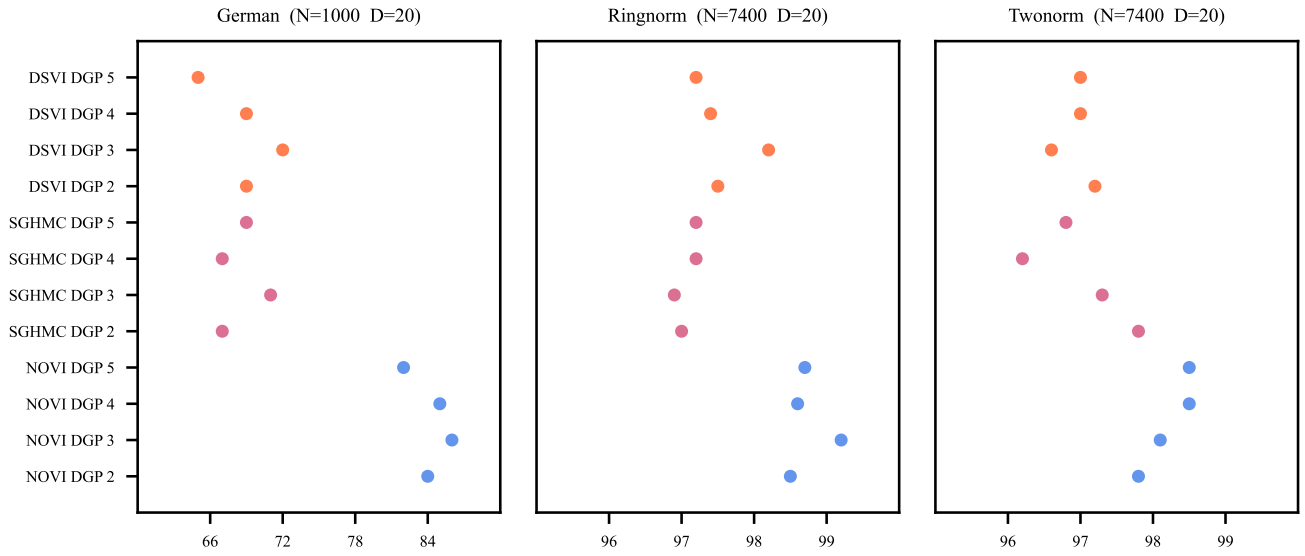

Fig. 3. Classification means test accuracy (%) by our NOVI method (blue), SGHMC (pink), and DSVI (orange) for DGPs on three UCI benchmark datasets. Higher is better.

Proof: From Cauchy–Schwarz inequality

$$\begin{aligned}\boldsymbol{a}^{\mathrm{T}}\boldsymbol{y}-\lambda\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}&\leqslant\|\boldsymbol{a}\|_2\|\boldsymbol{y}\|_2-\lambda\|\boldsymbol{y}\|_2^2\\&=\frac{1}{4\lambda}\|\boldsymbol{a}\|_2^2-\lambda\left(\|\boldsymbol{y}\|_2-\frac{1}{2\lambda}\|\boldsymbol{a}\|_2\right)^2\leqslant\frac{1}{4\lambda}\|\boldsymbol{a}\|_2^2.\end{aligned}\tag{38}$$

The equality holds iff $\boldsymbol{y}=(1/(2\lambda))\boldsymbol{a}$.

Definition 5: The Fisher divergence [60] between two suitably smooth density functions is defined as

$$\mathrm{FD}(q,\,p)=\int_{\mathbb{R}^d}\|\nabla\log q(\theta)-\nabla\log p(\theta)\|_2^2 q(\theta)d\theta.\tag{39}$$

Theorem 5: Training the generator with the optimal discriminator corresponds to minimizing the Fisher divergence between $q_{\boldsymbol{\theta}}$ and p, and the corresponding optimal loss for (12) is

$$\mathcal{L}(\boldsymbol{\theta},\,\boldsymbol{\nu})=\frac{1}{4\lambda}FD(q_{\boldsymbol{\theta}}(\mathcal{U}),\,p(\mathcal{U}|\mathcal{D},\,\boldsymbol{\nu}))\tag{40}$$

where $\lambda\in\mathbb{R}^+$ is a regularization strength defined in (9).

Proof: Let our loss function be $\mathcal{L}(\theta,\,\nu)$, by Lemma 3

$$\begin{aligned}\mathcal{L}(\theta,\,\nu)&=\sup_{\eta}\mathbb{E}_{q_\vartheta(\mathcal{U})}\left[\mathcal{A}_p\boldsymbol{\phi}_\eta(\mathcal{U})-\lambda\boldsymbol{\phi}_\eta(\mathcal{U})^{\mathrm{T}}\boldsymbol{\phi}_\eta(\mathcal{U})\right]\\&=\sup_{\eta}\mathbb{E}_{q_\vartheta(\mathcal{U})}\big[(\nabla_{\mathcal{U}}\log p(\mathcal{U}\mid\mathcal{D},\,\nu)\\&\quad-\nabla_{\mathcal{U}}q_\theta(\mathcal{U}))^{\mathrm{T}}\boldsymbol{\phi}_\eta(\mathcal{U})-\lambda\boldsymbol{\phi}_\eta(\mathcal{U})^{\mathrm{T}}\boldsymbol{\phi}_\eta(\mathcal{U})\big].\end{aligned}\tag{41}$$

According to Lemma 4, the above equation attains its maximum value when the function $\boldsymbol{\phi}_\eta(\mathcal{U})=\nabla_{\mathcal{U}}\log p(\mathcal{U}|\mathcal{D},\,\nu)-\nabla_{\mathcal{U}}q_\theta(\mathcal{U})$

$$\begin{aligned}\mathcal{L}(\theta,\,\nu)&=\frac{1}{4\lambda}\mathbb{E}_{q_\theta(\mathcal{U})}\left[\|\nabla_{\mathcal{U}}\log p(\mathcal{U}|\mathcal{D},\,\nu)-\nabla_{\mathcal{U}}q_\theta(\mathcal{U})\|_2^2\right]\\&=\frac{1}{4\lambda}FD(q_\theta(\mathcal{U}),\,p(\mathcal{U}|\mathcal{D},\,\nu)).\end{aligned}\tag{42}$$

The optimal discriminator is

$$\boldsymbol{\phi}_{\eta^\star}(\mathcal{U})=\frac{1}{2\lambda}(\nabla_{\mathcal{U}}\log p(\mathcal{U}|\mathcal{D},\,\nu)-\nabla_{\mathcal{U}}q_\theta(\mathcal{U})).\tag{43}$$

Lemma 5: Suppose $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ are probability densities on $\mathbb{R}^d$ and $\boldsymbol{\phi}\colon\mathbb{R}^d\to\mathbb{R}^d$ is a differentiable function that satisfies $\lim_{\|\boldsymbol{x}\|\to\infty}q(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x})=\boldsymbol{0}$, we have

$$\left|\mathbb{E}_{\boldsymbol{x}\sim q}\left[\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{x})\right]\right|\leqslant\sqrt{\mathbb{E}_{\boldsymbol{x}\sim q}\|\boldsymbol{\phi}(\boldsymbol{x})\|_2^2}\sqrt{FD(q,\,p)}.\tag{44}$$

Proof: By Lemma 3, we have

$$\begin{aligned}&\left|\mathbb{E}_{\boldsymbol{x}\sim q}\left[\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{x})\right]\right|\\&\quad=|\mathbb{E}_{\boldsymbol{x}\sim q}[(\nabla_{\boldsymbol{x}}\log p(x)-\nabla_{\boldsymbol{x}}\log q(x))^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})]|.\end{aligned}\tag{45}$$

From Cauchy–Schwarz inequality and Hölder's inequality

$$\begin{aligned}&|\mathbb{E}_{\boldsymbol{x}\sim q}[(\nabla_{\boldsymbol{x}}\log p(x)-\nabla_{\boldsymbol{x}}\log q(x))^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})]|\\&\leqslant\mathbb{E}_{\boldsymbol{x}\sim q}\left[\|\nabla_{\boldsymbol{x}}\log p(x)-\nabla_{\boldsymbol{x}}\log q(x)\|_2\|\boldsymbol{\phi}(\boldsymbol{x})\|_2\right]\\&\leqslant\sqrt{\mathbb{E}_{\boldsymbol{x}\sim q}\|\nabla_{\boldsymbol{x}}\log p(x)-\nabla_{\boldsymbol{x}}\log q(x)\|_2^2}\sqrt{\mathbb{E}_{\boldsymbol{x}\sim q}\|\boldsymbol{\phi}(\boldsymbol{x})\|_2^2}\end{aligned}$$

$$\begin{aligned}\|\nabla\log p(\mathbf{y}\mid\mathcal{U})-\nabla\log\tilde{p}(\mathbf{y}\mid\mathcal{U})\|&=\left\|\frac{\nabla p(\mathbf{y}\mid\mathcal{U})}{p(\mathbf{y}\mid\mathcal{U})}-\frac{\nabla\tilde{p}(\mathbf{y}\mid\mathcal{U})}{\tilde{p}(\mathbf{y}\mid\mathcal{U})}\right\|\\&=\left\|\frac{\tilde{p}(\mathbf{y}\mid\mathcal{U})\nabla p(\mathbf{y}\mid\mathcal{U})-p(\mathbf{y}\mid\mathcal{U})\nabla\tilde{p}(\mathbf{y}\mid\mathcal{U})}{p(\mathbf{y}\mid\mathcal{U})\tilde{p}(\mathbf{y}\mid\mathcal{U})}\right\|\\&=\left\|\frac{\tilde{p}(\mathbf{y}\mid\mathcal{U})\nabla p(\mathbf{y}\mid\mathcal{U})-p(\mathbf{y}\mid\mathcal{U})\nabla p(\mathbf{y}\mid\mathcal{U})+p(\mathbf{y}\mid\mathcal{U})\nabla p(\mathbf{y}\mid\mathcal{U})-p(\mathbf{y}\mid\mathcal{U})\nabla\tilde{p}(\mathbf{y}\mid\mathcal{U})}{p(\mathbf{y}\mid\mathcal{U})\tilde{p}(\mathbf{y}\mid\mathcal{U})}\right\|\\&=\left\|\frac{(\tilde{p}(\mathbf{y}\mid\mathcal{U})-p(\mathbf{y}\mid\mathcal{U}))\nabla p(\mathbf{y}\mid\mathcal{U})+p(\mathbf{y}\mid\mathcal{U})(\nabla p(\mathbf{y}\mid\mathcal{U})-\nabla\tilde{p}(\mathbf{y}\mid\mathcal{U}))}{p(\mathbf{y}\mid\mathcal{U})\tilde{p}(\mathbf{y}\mid\mathcal{U})}\right\|\\&\leqslant\frac{1}{p(\mathbf{y}\mid\mathcal{U})\tilde{p}(\mathbf{y}\mid\mathcal{U})}\cdot\begin{pmatrix}\|\nabla p(\mathbf{y}\mid\mathcal{U})\|\cdot\|\tilde{p}(\mathbf{y}\mid\mathcal{U})-p(\mathbf{y}\mid\mathcal{U})\|\\+p(\mathbf{y}\mid\mathcal{U})\|(\nabla p(\mathbf{y}\mid\mathcal{U})-\nabla\tilde{p}(\mathbf{y}\mid\mathcal{U}))\|\end{pmatrix}\xrightarrow{\mathcal{P}}0\end{aligned}\tag{27}$$

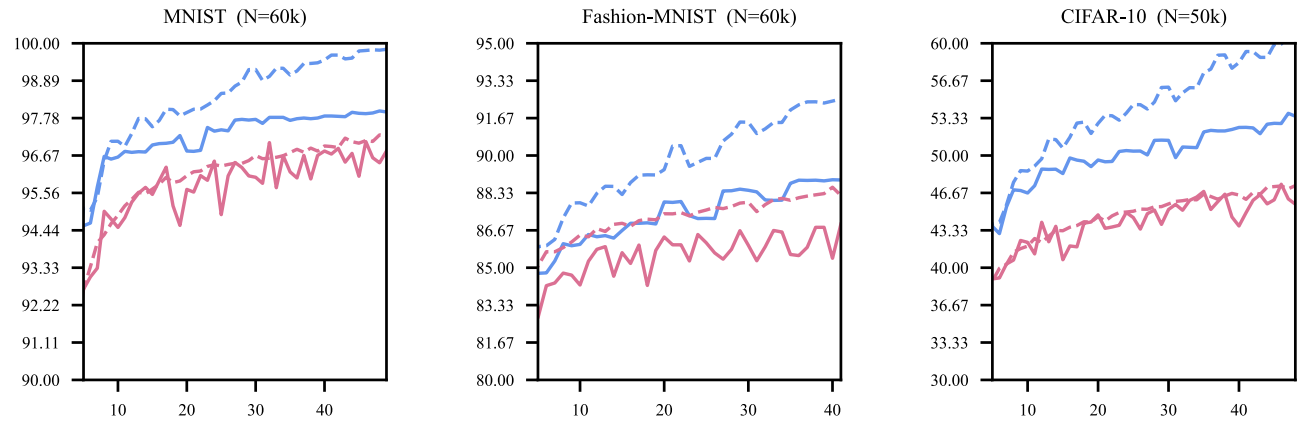

Fig. 4. Mean accuracy comparison of NOVI (blue) with Monto Carlo log-likelihood maximization method (pink) using three-layer DGP model on three image classification datasets. The results of the training and test sets are shown by dashed and solid lines, respectively.

$$= \sqrt{\mathbb{E}_{\boldsymbol{x}\sim q}\|\boldsymbol{\phi}(\boldsymbol{x})\|_2^2}\sqrt{FD(q,p)}. \tag{46}$$

Definition 6: Suppose $p(\boldsymbol{x})$ is probability densities on $\mathbb{R}^d$ and $\psi : \mathbb{R}^d \to \mathbb{R}$ is a function, we define $\boldsymbol{\phi}_{\psi}^{p}(\boldsymbol{x})$ as a solution of the Stein equation $\mathcal{A}_p\boldsymbol{\phi}(\boldsymbol{x}) = \psi(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}\sim p}[\psi(\boldsymbol{x})]$.

Lemma 6: Suppose $\psi : \mathbb{R}^d \to \mathbb{R}$ is a bounded function, there exists a bounded solution of the Stein equation.

Proof: Let $h(\boldsymbol{x}) = \psi(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}\sim p}[\psi(\boldsymbol{x})]$, $h(\boldsymbol{x})$ is obviously bounded, then

$$\begin{aligned}\phi_1(\boldsymbol{x}) &= \frac{1}{p(\boldsymbol{x})}\int_{-\infty}^{x_1} p(t, x_2, \ldots, x_d)h(t, x_2, \ldots, x_d)dt\\ \phi_2(\boldsymbol{x}) &= \cdots = \phi_d(\boldsymbol{x}) = 0\end{aligned} \tag{47}$$

is a bounded solution.

Lemma 7: Suppose $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ are probability densities on $\mathbb{R}^d$ and $\boldsymbol{\psi} : \mathbb{R}^d \to \mathbb{R}^n$ is a bounded function. $\forall i \in (1, \ldots, n)$, let $\boldsymbol{\phi}_{\psi_i}^{p}(\boldsymbol{x})$ be a solution of the Stein equation, then we have

$$\left\|\mathbb{E}_{\boldsymbol{x}\sim q}[\boldsymbol{\psi}(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim p}[\boldsymbol{\psi}(\boldsymbol{x})]\right\|_2 \leqslant c_{\boldsymbol{\psi}}^{p,q}\sqrt{\mathrm{FD}(q,p)} \tag{48}$$

where $c_{\boldsymbol{\psi}}^{p,q} \triangleq (\sum_{i=1}^{n}\mathbb{E}_{\boldsymbol{x}\sim q}\|\boldsymbol{\phi}_{\psi_i}^{p}(\boldsymbol{x})\|_2^2)^{1/2}$ is bounded.

Proof: By Lemma 5, we have

$$\begin{aligned}&\left|\mathbb{E}_{\boldsymbol{x}\sim q}[\psi_i(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim p}[\psi_i(\boldsymbol{x})]\right|\\ &= \left|\mathbb{E}_{\boldsymbol{x}\sim q}\left[\psi_i(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}\sim p}[\psi_i(\boldsymbol{x})]\right]\right|\\ &= \left|\mathbb{E}_{\boldsymbol{x}\sim q}\left[\mathcal{A}_p\boldsymbol{\phi}_{\psi_i}^{p}(\boldsymbol{x})\right]\right| \leqslant \sqrt{\mathbb{E}_{\boldsymbol{x}\sim q}\left\|\boldsymbol{\phi}_{\psi_i}^{p}(\boldsymbol{x})\right\|_2^2}\sqrt{\mathrm{FD}(q,p)}.\end{aligned} \tag{49}$$

As a result

$$\begin{aligned}&\left\|\mathbb{E}_{\boldsymbol{x}\sim q}\left[\boldsymbol{\psi}(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim p}\left[\boldsymbol{\psi}(\boldsymbol{x})\right]\right\|_2\\ &= \sqrt{\sum_{i=1}^{n}\left|\mathbb{E}_{\boldsymbol{x}\sim q}[\psi_i(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim p}[\psi_i(\boldsymbol{x})]\right|^2}\\ &\leqslant \sqrt{\sum_{i=1}^{n}\mathbb{E}_{\boldsymbol{x}\sim q}\left\|\boldsymbol{\phi}_{x_i}^{p}(\boldsymbol{x})\right\|_2^2\mathrm{FD}(q,p)} = c_{\boldsymbol{\psi}}^{p,q}\sqrt{\mathrm{FD}(q,p)}\end{aligned} \tag{50}$$

where

$$c_{\boldsymbol{\psi}}^{p,q} \triangleq \sqrt{\sum_{i=1}^{n}\mathbb{E}_{\boldsymbol{x}\sim q}\left\|\boldsymbol{\phi}_{\psi_i}^{p}(\boldsymbol{x})\right\|_2^2} \leqslant \sqrt{\sum_{i=1}^{n}\left\|\boldsymbol{\phi}_{\psi_i}^{p}(\boldsymbol{x})\right\|_{\infty}^2} \tag{51}$$

is bounded by Lemma 6.

Theorem 6: The bias of the estimate of the prediction $\widehat{\mathbf{F}}_L^{\star}$ in (21) from the DGPs exact evaluation can be bounded by the square root of the Fisher divergence between $q_{\theta}(\mathcal{U})$ and $p(\mathcal{U}|\mathcal{D}, \nu)$ up to multiplying a constant.

Proof: From the Law of Large Numbers, we have

$$\hat{\mathbf{F}}_L^{\star} = \frac{1}{S}\sum_{s=1}^{S}\hat{\mathbf{F}}_L^{\star(s)} \approx \mathbb{E}_{q(\mathbf{F}_L^{\star})}\left[\mathbf{F}_L^{\star}\right] \tag{52}$$

where S denotes the number of samples involved in the estimation and $q(\mathbf{F}_L^{\star})$ is represented as

$$q(\mathbf{F}_L^{\star}) = \int\prod_{\ell=1}^{L}\prod_{d=1}^{D_\ell} p(\mathbf{F}_{\ell,d}^{\star}|\mathbf{F}_{\ell-1}^{\star}, \mathbf{U}_{\ell,d})q_{\theta^{\star}}(\mathbf{U}_{\ell})d\mathbf{F}_{\ell-1}^{\star}d\mathbf{U}_{\ell,d}. \tag{53}$$

The DGP exact evaluation can be written as

$$\widetilde{\mathbf{F}}_L^{\star} = \mathbb{E}_{p(\mathbf{F}_L^{\star}|\mathcal{D},\nu)}[\mathbf{F}_L^{*}]. \tag{54}$$

Similarly

$$\begin{aligned}&p(\mathbf{F}_L^{\star}|\mathcal{D}, \nu)\\ &= \int\prod_{\ell=1}^{L}\prod_{d=1}^{D\ell} p(\mathbf{F}_{\ell}^{\star} \mid \mathbf{F}_{\ell-1}^{\star}, \mathbf{U}_{\ell,d})\,p(\mathbf{U}_{\ell}|\mathcal{D}, \nu)d\mathbf{F}_{\ell-1}^{\star}d\mathbf{U}_{\ell,d}.\end{aligned} \tag{55}$$

By Lemma 7

$$\begin{aligned}&\left\|\widehat{\mathbf{F}}_L^{\star} - \widetilde{\mathbf{F}}_L^{\star}\right\|_2\\ &= \left\|\mathbb{E}_{q(\mathbf{F}_L^{\star})}[\mathbf{F}_L^{\star}] - \mathbb{E}_{p(\mathbf{F}_L^{\star}|\mathcal{D},\nu)}\left[\mathbf{F}_L^{\star}\right]\right\|_2\\ &= \left\|\mathbb{E}_{q(\mathcal{U})}\left[\int\mathbf{F}_L^{\star}\prod_{\ell=1}^{L}\prod_{d=1}^{D_\ell} p(\mathbf{F}_{\ell,d}^{\star}|\mathbf{F}_{\ell-1}^{\star}, \mathbf{U}_{\ell,d})d\mathbf{F}_{\ell-1}^{\star}d\mathbf{F}_L^{\star}\right]\right.\\ &\quad\left. -\mathbb{E}_{p(\mathcal{U}|\mathcal{D},\nu)}\left[\int\mathbf{F}_L^{\star}\prod_{\ell=1}^{L}\prod_{d=1}^{D_\ell} p(\mathbf{F}_{\ell,d}^{\star}|\mathbf{F}_{\ell-1}^{\star}, \mathbf{U}_{\ell,d})d\mathbf{F}_{\ell-1}^{\star}d\mathbf{F}_L^{\star}\right]\right\|_2\\ &= \left\|\mathbb{E}_{q(\mathcal{U})}\left[\boldsymbol{\psi}(\mathcal{U})\right] - \mathbb{E}_{p(\mathcal{U}|\mathcal{D},\nu)}\left[\boldsymbol{\psi}(\mathcal{U})\right]\right\|_2\\ &\leqslant c_{\boldsymbol{\psi}}^{p,q}\sqrt{FD(q(\mathcal{U}), p(\mathcal{U}|\mathcal{D}, \nu))}.\end{aligned} \tag{56}$$

Since Ω and Υ are both compact, $\boldsymbol{\psi}(\mathcal{U}) = \int\mathbf{F}_L^{\star}\prod_{\ell=1}^{L}\prod_{d=1}^{D_\ell} p(\mathbf{F}_{\ell,d}^{\star}|\mathbf{F}_{\ell-1}^{\star}, \mathbf{U}_{\ell,d})d\mathbf{F}_{\ell-1}^{\star}d\mathbf{F}_L^{\star}$ is obviously bounded.

APPENDIX D
ADDITIONAL RESULTS

A. UCI Classification Benchmark

We performed classification tasks on three UCI benchmark datasets, with sizes ranging from 1000 to 7400. Results are

TABLE V

TABULAR VERSION OF FIG. 2 IN THE MAIN TEXT

Data	DSVI 2	DSVI 3	DSVI 4	DSVI 5	SGHMC 2	SGHMC 3	SGHMC 4	SGHMC 5	IPVI 2	IPVI 3	IPVI 4	IPVI 5	NOVI 2	NOVI 3	NOVI 4	NOVI 5
Boston	0.32 (0.02)	0.32 (0.02)	0.32 (0.02)	0.32 (0.02)	0.37 (0.07)	0.38 (0.08)	0.35 (0.09)	0.39 (0.07)	0.35 (0.06)	0.34 (0.05)	0.33 (0.06)	0.32 (0.04)	**0.20 (0.01)**	0.34 (0.02)	0.40 (0.03)	0.38 (0.02)
Energy	0.05 (0.00)	0.05 (0.00)	0.05 (0.00)	0.05 (0.00)	0.13 (0.01)	0.13 (0.01)	0.09 (0.04)	0.13 (0.01)	0.13 (0.01)	0.13 (0.01)	0.12 (0.03)	0.11 (0.04)	**0.04 (0.00)**	0.05 (0.00)	0.06 (0.00)	0.06 (0.00)
Power	0.22 (0.00)	0.22 (0.00)	0.22 (0.00)	0.22 (0.00)	0.23 (0.00)	0.22 (0.00)	0.22 (0.00)	0.22 (0.00)	0.22 (0.01)	0.22 (0.01)	0.22 (0.01)	0.21 (0.01)	0.22 (0.00)	**0.21 (0.00)**	**0.21 (0.00)**	**0.21 (0.00)**
Concrete	0.34 (0.01)	0.34 (0.01)	0.34 (0.01)	0.34 (0.01)	0.35 (0.03)	0.33 (0.03)	0.31 (0.02)	0.31 (0.02)	0.32 (0.02)	0.30 (0.03)	0.31 (0.03)	0.30 (0.04)	0.24 (0.00)	0.25 (0.00)	0.24 (0.00)	**0.23 (0.00)**
Yacht	0.07 (0.00)	0.07 (0.00)	0.07 (0.00)	0.07 (0.00)	0.03 (0.01)	0.03 (0.01)	**0.02 (0.01)**	0.03 (0.01)	0.03 (0.02)	0.03 (0.02)	0.04 (0.03)	0.03 (0.01)	0.03 (0.00)	0.09 (0.01)	0.08 (0.00)	0.06 (0.00)
Qsar	0.57 (0.00)	0.50 (0.00)	0.47 (0.00)	**0.42 (0.00)**	0.56 (0.00)	0.56 (0.00)	0.56 (0.00)	0.56 (0.00)	0.56 (0.01)	0.54 (0.01)	0.54 (0.01)	0.54 (0.01)	0.51 (0.00)	0.46 (0.01)	0.45 (0.01)	0.44 (0.01)
Protein	0.81 (0.00)	0.77 (0.00)	0.79 (0.00)	0.73 (0.00)	0.72 (0.01)	0.71 (0.01)	0.70 (0.01)	0.69 (0.00)	0.68 (0.01)	0.65 (0.01)	0.65 (0.01)	0.65 (0.01)	0.67 (0.00)	**0.65 (0.00)**	0.66 (0.00)	0.66 (0.00)
Kin8nm	0.39 (0.00)	0.37 (0.00)	0.34 (0.00)	0.30 (0.00)	0.26 (0.01)	0.25 (0.01)	0.25 (0.01)	0.24 (0.01)	0.25 (0.01)	0.25 (0.01)	0.25 (0.00)	0.26 (0.01)	**0.24 (0.00)**	0.28 (0.00)	0.26 (0.00)	0.27 (0.00)

TABLE VI

COMPARISON OF NUMBER OF INDUCING POINTS (50, 100, 200, AND 400) USING TWO-LAYER DGP MODEL ON FOUR UCI REGRESSION DATASETS. M DENOTES THE NUMBER OF INDUCING POINTS PER LAYER

	Concrete	**Energy**	**Boston**	**Kin8nm**
Iteration	500	600	300	500
RMSE (M=50)	0.28 (0.00)	0.04 (0.00)	0.23 (0.00)	0.26 (0.00)
Time (M=50)	0.397s	0.404s	0.380s	0.600s
RMSE (M=100)	0.24 (0.00)	0.04 (0.00)	0.20 (0.01)	0.24 (0.00)
Time (M=100)	0.403s	0.420s	0.400s	0.613s
RMSE (M=200)	0.20 (0.00)	0.03 (0.00)	0.20 (0.00)	0.24 (0.00)
Time (M=200)	0.408s	0.450s	0.410s	0.646s
RMSE (M=400)	0.19 (0.00)	0.03 (0.00)	0.18 (0.01)	0.23 (0.00)
Time (M=400)	0.408s	0.450s	0.420s	0.658s

reported in Fig. 3 compared through test accuracy as the performance metric. It can be observed that NOVI achieves the best results in different sizes of datasets and shows competitive performance within different layers.

B. Ablation Study on Classification Datasets

We also performed an ablation study on classification datasets and reported its results by test accuracy in Fig. 4. From this it can be seen that NOVI not only achieves better results on large-scale datasets, which demonstrates its scalability, but also the results on the test set have far exceeded the performance of the Monte Carlo log-likelihood maximization method on the training set, suggesting the feasibility of adversarial training.

C. Tabular Version of Fig. 1 in the Main Text

The tabular version of Fig. 1 in the main text can be seen in Table V.

D. Comparison About Inducing Points

In order to investigate the robustness of NOVI at different numbers of induced points, we have performed an ablation study to compare accuracy and training time on four UCI regression datasets using a two-layer DGP model. For each dataset, the number of iterations is set to be the same for fair comparison. Results are shown in Table VI. From this, it can be seen that the performance increases gradually with the number of induction points, while the time fluctuates only slightly, which shows the robustness of NOVI to the number of inducing points.

E. Additional Experiments

We have performed additional regression experiments for two real-world datasets: Estate and Elevators. Results are shown in Table VII. From these two datasets, it can be seen that NOVI has achieved a better RMSE value than the other two methods.

In order to further demonstrate the advantages of our method compared to the SOTA approaches, we conducted comparisons with the two most recent methods for DGP posterior inference, IWVI [78] and its variant (IWVI with DREG estimators) [79]. Introducing importance weighting for posterior sampling in IWVI not only enhances the variational lower bound but also serves as a crucial variation of DSVI. In contrast, our proposed method, as explained in the text, distinguishes itself from variational approaches based on KL divergence. We presented their results on eight UCI datasets in Table VIII. The results show that NOVI still outperforms the latest DGP posterior inference methods.

APPENDIX E
TRAINING DETAILS

A. UCI Datasets

1) Training: We conducted a random 0.9/0.1 train/test split and normalized the features to the range $[-1, 1]$. The depth L of DGP models varied from 2 to 5, with 100 inducing points per layer, which are initialized by sampling from isotropic Gaussian distribution. The output dimension for each hidden layer is set to 1 for the final layer and 10 for others. We have utilized the RQ kernel for all tasks. For all datasets, we have optimized hyper-parameters and network parameters jointly and utilized different learning rates, 0.02 for hyper-parameters and 0.001 for network parameters using Adam optimizer [80]. The dimension of noise ϵ used to generate $\mathcal{U}$ is set to 200 for all datasets. We train for almost 500 iterations for all datasets. DSVI and SGHMC methods are initialized the same as NOVI to obtain a fair comparison.

2) Network Settings: In this study, the selection of the generator and discriminator networks is done manually. However, to further optimize the hyperparameters of these neural networks, we propose a classical grid search approach for each experimental dataset. Taking the energy dataset as an example, we set the generator and discriminator networks as three-layer neural networks. To explore the optimal choices for the activation function, we present the results in Table IX. From the results in Table IX, we can identify the relatively superior combination from the alternative choices of generator and discriminator networks. For the other hyperparameters of the neural networks, such as the number of hidden units in the intermediate layers, we can also fine-tune them using the same grid search method. As for further improvements or refinements of the algorithm, we leave it as future work. By adopting this systematic grid search method, we can effectively optimize the hyperparameters of the generator and

TABLE VII

ADDITIONAL EXPERIMENTS FOR REAL-WORLD DATASETS. IT SHOWS REGRESSION MEAN TEST RMSE VALUES WITH ITS STANDARD DEVIATION ON THE ROUND BRACKET. L DENOTES THE NUMBER OF LAYERS IN DGP MODELS

Method	Estate				Elevators			
	L=2	L=3	L=4	L=5	L=2	L=3	L=4	L=5
DSVI	0.65 (0.02)	0.66 (0.02)	0.50 (0.02)	0.64 (0.02)	0.37 (0.00)	0.36 (0.00)	0.37 (0.00)	0.36 (0.00)
SGHMC	0.54 (0.01)	0.50 (0.01)	0.53 (0.01)	0.61 (0.01)	0.36 (0.00)	0.36 (0.00)	**0.35 (0.00)**	**0.35 (0.00)**
NOVI	0.56 (0.02)	0.40 (0.02)	0.40 (0.01)	**0.39 (0.02)**	0.36 (0.00)	**0.35 (0.00)**	**0.35 (0.00)**	**0.35 (0.00)**

TABLE VIII

COMPARISONS WITH THE TWO MOST RECENT METHODS; IWVI AND IWVI WITH DREG ESTIMATORS. THE RESULTS OF UCI TEST RMSE ARE REPORTED

Data	IWVI 2	IWVI 3	IWVI 4	IWVI 5	IWVI-DREG 2	IWVI-DREG 3	IWVI-DREG4	IWVI-DREG 5	NOVI 2	NOVI 3	NOVI 4	NOVI 5
Boston	0.33 (0.02)	0.35 (0.02)	0.35 (0.02)	0.36 (0.02)	0.32 (0.02)	0.33 (0.02)	0.35 (0.02)	0.36 (0.02)	**0.20 (0.01)**	0.34 (0.02)	0.40 (0.03)	0.38 (0.02)
Energy	0.05 (0.00)	0.06 (0.00)	0.06 (0.00)	0.06 (0.00)	0.05 (0.00)	0.06 (0.00)	0.06 (0.00)	0.06 (0.00)	**0.04 (0.00)**	0.05 (0.00)	0.06 (0.00)	0.06 (0.00)
Power	0.22 (0.00)	0.22 (0.00)	0.22 (0.00)	0.22 (0.00)	0.22 (0.00)	0.22 (0.00)	0.22 (0.00)	0.22 (0.00)	0.22 (0.00)	**0.21 (0.00)**	**0.21 (0.00)**	**0.21 (0.00)**
Concrete	0.32 (0.01)	0.28 (0.01)	0.27 (0.01)	0.27 (0.01)	0.31 (0.01)	0.27 (0.01)	0.27 (0.01)	0.26 (0.01)	0.24 (0.00)	0.25 (0.00)	0.24 (0.00)	**0.23 (0.00)**
Yacht	0.07 (0.00)	0.09 (0.00)	0.07 (0.00)	0.06 (0.00)	0.07 (0.00)	0.08 (0.00)	0.07 (0.0)	0.06 (0.00)	**0.03** (0.00)	0.09 (0.01)	0.08 (0.00)	0.06 (0.00)
Qsar	0.54 (0.00)	0.47 (0.00)	0.44 (0.00)	0.43 (0.00)	0.52 (0.00)	0.46 (0.00)	0.43 (0.00)	**0.42 (0.00)**	0.51 (0.00)	0.46 (0.01)	0.45 (0.01)	0.44 (0.01)
Protein	0.72 (0.00)	0.67 (0.00)	0.67 (0.00)	0.67 (0.00)	0.70 (0.01)	0.66 (0.01)	0.66 (0.01)	0.66 (0.00)	0.67 (0.00)	**0.65 (0.00)**	0.66 (0.00)	0.66 (0.00)
Kin8nm	0.37 (0.00)	0.34 (0.00)	0.31 (0.00)	0.29 (0.00)	0.35 (0.00)	0.32 (0.00)	0.31 (0.00)	0.29 (0.00)	**0.24 (0.00)**	0.28 (0.00)	0.26 (0.00)	0.27 (0.00)

TABLE IX

EXAMPLE: WHEN BOTH THE GENERATOR AND DISCRIMINATOR ARE THREE-LAYER NEURAL NETWORKS, AND THEIR ACTIVATION FUNCTIONS ARE, RESPECTIVELY, TAKEN AS TANH, PRELU, AND SIGMOID, THE RMSE OF THE NOVI ALGORITHM ON THE ENERGY DATASET IS REPORTED

Discriminator \ Generator	Tanh	Prelu	Sigmoid
Tanh	0.041(0.001)	0.042(0.001)	0.039(0.003)
Prelu	0.041(0.001)	0.038(0.001)	0.040 (0.003)
Sigmoid	0.038(0.001)	0.034(0.003)	0.040(0.002)

TABLE X

EXAMPLE: THE RMSE VALUES OF THE TRAINING AND TESTING SETS CORRESPONDING TO DIFFERENT λ VALUES ON THE ENERGY DATASET

Dataset \ λ	1	10	100	1000	10000
Train	0.025(0.001)	0.031(0.001)	0.032(0.001)	0.042(0.001)	0.050(0.001)
Test	0.050(0.001)	0.037(0.001)	0.038 (0.003)	0.043 (0.003)	0.050 (0.003)

TABLE XI

IMPACT OF UTILIZING THE HUTCHINSON ESTIMATOR FOR CALCULATING THE TRACE OF THE JACOBIAN MATRIX ON THE COMPUTATIONAL COMPLEXITY

	Hutchinson Estimator	Direct Computation
Energy	0.391s/iter	0.454s/iter
Elevators	0.492s/iter	0.587s/iter

discriminator networks, leading to improved performance on the specific dataset under consideration. This approach offers a structured framework for selecting and fine-tuning the neural network hyperparameters in the context of our study.

3) Regularization Strategies: Based on our experimental results on the energy dataset, we tested the effect of different λ values on the model performance. We took λ values in increments of 10, namely, 1, 10, 100, 1000, and 10 000, maintained the other hyperparameters unchanged, and recorded the experimental results (RMSE on both the training and testing sets) in Table X. From Table X, it can be observed that the model performs the best when lambda is set to 10. Larger lambda values may lead to over-regularization, thereby reducing the model performance on the training set. On the other hand, too small lambda values may result in overfitting, although the model performs better on the training set, it fails to generalize on the testing set. Therefore, it is recommended to select an appropriate lambda value to adjust the regularization degree of the model. By conducting experiments on different λ values, we can find the optimal lambda value that achieves good performance on both the training and testing sets.

In order to investigate the impact of using the Hutchinson estimator on NOVI, as proposed in our main text to significantly reduce the computational complexity of calculating the trace of the Jacobian matrix, we conducted an ablation experiment on the Energy dataset and a considerably large Elevators dataset consisting of over ten thousand data points and 18-D features. We compared the direct computation of the Jacobian matrix with the iterative approach using the Hutchinson estimator and reported the corresponding time consumption in Table XI. From the results presented in the table, it can be observed that the Hutchinson estimator significantly reduces the computational complexity of the NOVI method, which aligns with the theoretical predictions.

B. Image Datasets

1) Training: We have followed the division of the original dataset and normalized pixel values to $[-1, 1]$. The depth L of DGP models vary from 3 to 4 with 100 inducing points per layer, which are initialized by sampling from isotropic Gaussian distribution. The output dimension for each hidden layer is set to be 10 for the final layer (which is the exact number of classes to predict), and 60 for others. We have utilized the RQ kernel for all tasks. For all datasets, we have optimized hyper-parameters and network parameters jointly and utilized different learning rates, 0.02 for hyper-parameters and 0.001 for network parameters using Adam optimizer [80]. The dimension of noise $\boldsymbol{\epsilon}$ used to generate $\mathcal{U}$ is set to 200 for all datasets. We train for almost 10K iterations for all datasets. DSVI and SGHMC methods are initialized the same as NOVI to obtain a fair comparison.

2) Network Settings: The selection of the generator and discriminator networks are done manually, we also use a classical grid search approach for each experimental dataset.

REFERENCES

[1] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[2] J. Cheng, Y. Chen, Q. Zhang, L. Gan, C. Liu, and M. Liu, "Real-time trajectory planning for autonomous driving with Gaussian process and incremental refinement," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 8999–9005.

[3] G. Chowdhary, H. A. Kingravi, J. P. How, and P. A. Vela, "Bayesian nonparametric adaptive control using Gaussian processes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 537–550, Mar. 2015.

[4] W. Cho, Y. Kim, and J. Park, "Hierarchical anomaly detection using a multioutput Gaussian process," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 261–272, Jan. 2020.

[5] V. Dutordoir, J. Hensman, M. van der Wilk, C. H. Ek, Z. Ghahramani, and N. Durrande, "Deep neural networks as point estimates for deep Gaussian processes," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 9443–9455.

[6] A. Hebbal, L. Brevault, M. Balesdent, E.-G. Taibi, and N. Melab, "Efficient global optimization using deep Gaussian processes," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8.

[7] C.-K. Lu, S. C.-H. Yang, X. Hao, and P. Shafto, "Interpretable deep Gaussian processes with moments," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 613–623.

[8] S. W. Ober and L. Aitchison, "Global inducing point variational posteriors for Bayesian neural networks and deep Gaussian processes," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8248–8259.

[9] M. M. Dunlop, M. A. Girolami, A. M. Stuart, and A. L. Teckentrup, "How deep are deep Gaussian processes?" *J. Mach. Learn. Res.*, vol. 19, no. 54, pp. 1–46, 2018.

[10] E. L. Snelson and Z. Gharahmani, "Sparse Gaussian processes using pseudo-inputs," in *Proc. Conf. Neural Inf. Process. Syst.*, 2005, pp. 1257–1264.

[11] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *J. Mach. Learn. Res.*, vol. 6, pp. 1939–1959, Dec. 2005.

[12] D. J. C. Mackay and M. N. Gibbs, "Variational Gaussian process classifiers," *IEEE Trans. Neural Netw.*, vol. 11, no. 6, pp. 1458–1464, Jan. 2000.

[13] L. Mao and S. Sun, "Multiview variational sparse Gaussian processes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 2875–2885, Jul. 2021.

[14] J. Hensman, A. Matthews, and Z. Ghahramani, "Scalable variational Gaussian process classification," in *Proc. 18th Int. Conf. Artif. Intell. Statist.*, vol. 38, G. Lebanon and S. V. N. Vishwanathan, Eds. San Diego, CA, USA, May 2015, pp. 351–360. [Online]. Available: https://proceedings.mlr.press/v38/hensman15.html

[15] M. P. Deisenroth and J. W. Ng, "Distributed Gaussian processes," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1481–1490.

[16] Y. Gal, M. van der Wilk, and C. E. Rasmussen, "Distributed variational inference in sparse Gaussian process regression and latent variable models," in *Proc. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3257–3265.

[17] J. Hensman, N. Fusi, and N. D. Lawrence, "Gaussian processes for big data," in *Proc. 29th Conf. Uncertainty Artif. Intell.*, 2013, pp. 282–290.

[18] T. N. Hoang, Q. M. Hoang, and B. K. H. Low, "A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 569–578.

[19] T. N. Hoang, Q. M. Hoang, and B. K. H. Low, "A distributed variational inference framework for unifying parallel sparse Gaussian process regression models," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 382–391.

[20] M. K. Titsias, "Variational model selection for sparse Gaussian process regression," School Comput. Sci., Univ. Manchester, Manchester, U.K., Tech. Rep. 1, 2009.

[21] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When Gaussian process meets big data: A review of scalable GPs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4405–4423, Nov. 2020.

[22] H. Salimbeni and M. Deisenroth, "Doubly stochastic variational inference for deep Gaussian processes," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4588–4599.

[23] M. Havasi, J. M. Hernández-Lobato, and J. J. Murillo-Fuentes, "Inference in deep Gaussian processes using stochastic gradient Hamiltonian Monte Carlo," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018, pp. 7517–7527.

[24] X. Gao, M. Gürbüzbalaban, and L. Zhu, "Global convergence of stochastic gradient Hamiltonian Monte Carlo for nonconvex stochastic optimization: Nonasymptotic performance bounds and momentum-based acceleration," *Oper. Res.*, vol. 70, no. 5, pp. 2931–2947, Sep. 2022.

[25] L. Wu, A. Miller, L. Anderson, G. Pleiss, D. Blei, and J. Cunningham, "Hierarchical inducing point Gaussian process for inter-domian observations," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 2926–2934.

[26] J. Lindinger, D. Reeb, C. Lippert, and B. Rakitsch, "Beyond the mean-field: Structured deep Gaussian processes improve the predictive uncertainties," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 8498–8509.

[27] J. Shi, M. Titsias, and A. Mnih, "Sparse orthogonal variational inference for Gaussian processes," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 1932–1942.

[28] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 37, 2015, pp. 1530–1538.

[29] H. Yu, D. Liu, B. K. H. Low, and P. Jaillet, "Convolutional normalizing flows for deep Gaussian processes," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–6.

[30] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," 2016, *arXiv:1605.08803*.

[31] L. Mescheder, S. Nowozin, and A. Geiger, "Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2017, pp. 2391–2400.

[32] C. Ma, Y. Li, and J. M. Hernández-Lobato, "Variational implicit processes," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4222–4233.

[33] H. Yu, Y. Chen, B. K. H. Low, P. Jaillet, and Z. Dai, "Implicit posterior variational inference for deep Gaussian processes," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[34] S. Sun, G. Zhang, J. Shi, and R. Grosse, "Functional variational Bayesian neural networks," 2019, *arXiv:1903.05779*.

[35] S. Rodríguez-Santana and D. Hernández-Lobato, "Adversarial α-divergence minimization for Bayesian approximate inference," *Neurocomputing*, vol. 471, pp. 260–274, Jan. 2022.

[36] S. Rodriguez-Santana, B. Zaldivar, and D. Hernandez-Lobato, "Function-space inference with sparse implicit processes," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 18723–18740.

[37] M. Sugiyama, T. Suzuki, and T. Kanamori, *Density Ratio Estimation in Machine Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[38] M. K. Titsias and F. J. R. Ruiz, "Unbiased implicit variational inference," in *Proc. Conf. Artif. Intell. Statist.*, 2019, pp. 167–176.

[39] Q. Liu and D. Wang, "Stein variational gradient descent: A general purpose Bayesian inference algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.

[40] D. Wang, X. Qin, F. Song, and L. Cheng, "Stabilizing training of generative adversarial nets via Langevin stein variational gradient descent," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 7, pp. 2768–2780, Jul. 2020.

[41] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[42] J. H. Huggins, T. Campbell, M. Kasprzak, and T. Broderick, "Practical bounds on the error of Bayesian posterior approximations: A nonasymptotic approach," 2018, *arXiv:1809.09505*.

[43] M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, 2009, pp. 567–574.

[44] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *J. Mach. Learn. Res.*, vol. 14, pp. 1303–1347, May 2013.

[45] A. Damianou and N. Lawrence, "Deep Gaussian processes," in *Proc. Conf. Artif. Intell. Statist.*, 2013, pp. 207–215.

[46] M. Opper and D. Saad, *Advanced Mean Field Methods: Theory and Practice*. Cambridge, MA, USA: MIT Press, 2001.

[47] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent.*, 2013, pp. 1–14.

[48] R. Ranganath, D. Tran, J. Altosaar, and D. Blei, "Operator variational inference," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.

[49] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, and R. Zemel, "Learning the Stein discrepancy for training and evaluating energy-based models without sampling," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2020, pp. 3732–3747.

[50] F. Huszár, "Variational inference using implicit distributions," 2017, *arXiv:1702.08235*.

[51] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control, Signals, Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989.

[52] Y. Lu and J. Lu, "A universal approximation theorem of deep neural networks for expressing probability distributions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 3094–3105.

[53] D. Perekrestenko, S. Müller, and H. Bölcskei, "Constructive universal high-dimensional distribution generation through deep ReLU networks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 7610–7619.
[54] Y. Yang, Z. Li, and Y. Wang, "On the capacity of deep generative networks for approximating distributions," *Neural Netw.*, vol. 145, pp. 144–154, Jan. 2022.
[55] L. Yang and G. E. Karniadakis, "Potential flow generator with L2 optimal transport regularity for generative models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 528–538, Feb. 2022.
[56] Q. Xie, P. Zhang, B. Yu, and J. Choi, "Semisupervised training of deep generative models for high-dimensional anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2444–2453, Jun. 2022.
[57] M. F. Hutchinson, "A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines," *Commun. Statist. Simul. Comput.*, vol. 18, no. 3, pp. 1059–1076, Jan. 1989.
[58] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 1278–1286.
[59] A. Blum, N. Haghtalab, and A. D. Procaccia, "Variational dropout and the local reparameterization trick," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 2575–2583.
[60] B. Sriperumbudur, K. Fukumizu, A. Gretton, A. Hyvärinen, and R. Kumar, "Density estimation in infinite dimensional exponential families," *J. Mach. Learn. Res.*, vol. 18, no. 57, pp. 1–59, 2017.
[61] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. NIPS*, vol. 33. Vancouver, BC, Canada: Curran Associates, 2020, pp. 6840–6851.
[62] M. Nikolova, "An algorithm for total variation minimization and applications," *J. Math. Imag. Vis.*, vol. 20, no. 1, pp. 89–97, Jan. 2004.
[63] R. Beran, "Minimum Hellinger distance estimates for parametric models," *Ann. Statist.*, vol. 5, no. 3, pp. 445–463, May 1977.
[64] S. S. Vallender, "Calculation of the Wasserstein distance between probability distributions on the line," *Theory Probab. Appl.*, vol. 18, no. 4, pp. 784–786, Sep. 1974.
[65] C. Ley and Y. Swan, "Stein's density approach and information inequalities," *Electron. Commun. Probab.*, vol. 18, pp. 1–14, Jan. 2013.
[66] W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud, "FFJORD: Free-form continuous dynamics for scalable reversible generative models," 2018, *arXiv:1810.01367*.
[67] A. Tsitsulin et al., "The shape of data: Intrinsic distance for data distributions," 2019, *arXiv:1905.11141*.
[68] I. Han, D. Malioutov, H. Avron, and J. Shin, "Approximating spectral sums of large-scale matrices using stochastic Chebyshev approximations," *SIAM J. Sci. Comput.*, vol. 39, no. 4, pp. A1558–A1585, Jan. 2017.
[69] V. Kumar, V. Singh, P. K. Srijith, and A. Damianou, "Deep Gaussian processes with convolutional kernels," 2018, *arXiv:1806.01655*.
[70] Y. LeCun, C. Cortes, and C. Burges. (1998). *MNIST Handwritten Digit Database*. [Online]. Available: http://www.research.att.com/~yann/ocr/mnist
[71] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[72] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 1, 2009.
[73] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[74] A. G. Wilson, Z. Hu, R. R. Salakhutdinov, and E. P. Xing, "Stochastic variational deep kernel learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.
[75] V. Dutordoir, M. Wilk, A. Artemev, and J. Hensman, "Bayesian image classification with deep convolutional Gaussian processes," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 1529–1539.
[76] K. Blomqvist, S. Kaski, and M. Heinonen, "Deep convolutional Gaussian processes," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, Würzburg, Germany. Cham, Switzerland: Springer, Sep. 2019, pp. 582–597.
[77] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 1997–2017, 2019.
[78] H. Salimbeni, V. Dutordoir, J. Hensman, and M. Deisenroth, "Deep Gaussian processes with importance-weighted variational inference," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5589–5598.
[79] T. G. Rudner, O. Key, Y. Gal, and T. Rainforth, "On signal-to-noise ratio issues in variational inference for deep Gaussian processes," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 9148–9156.
[80] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

Jian Xu received the B.S. degree from the Department of Business Administration, Communication University of China, Beijing, China, in 2017. He is currently pursuing the Ph.D. degree with the South China University of Technology, Guangzhou, China, focusing on machine learning, stochastic processes, generative models, and their applications.

Shian Du received the B.S. degree in applied mathematics from the South China University of Technology, Guangzhou, China, in 2023. He is currently pursuing the master's degree with Tsinghua University, Shenzhen, China.

His research interests include large-scale text-to-video and image-to-video generation.

Junmei Yang received the M.S. degree in cybernetics from the Chinese Academy of Sciences, Beijing, China, in 2005, and the Ph.D. degree in systems science from the Graduate School of Informatics, Kyoto University, Kyoto, Japan, in 2008.

She is currently an Associate Professor with the South China University of Technology, Guangzhou, China. Her current research interests focus on image processing, speech enhancement, machine learning, and artificial intelligence.

Qianli Ma (Member, IEEE) received the Ph.D. degree in computer science from the South China University of Technology, Guangzhou, China, in 2008.

From 2016 to 2017, he was a Visiting Scholar with the University of California San Diego, San Diego, CA, USA. He is currently a Professor with the School of Computer Science and Engineering, South China University of Technology. His current research interests include machine learning algorithms, data-mining methodologies, and their applications.

Delu Zeng (Member, IEEE) received the bachelor's degree in applied mathematics and the Ph.D. degree in information and signal processing from the South China University of Technology (SCUT), Guangzhou, China, in June 2003 and June 2010, respectively.

He was a Visiting Scholar with Columbia University, New York City, NY, USA, the University of Oulu, Oulu, Finland, and the University of Waterloo, Waterloo, ON, Canada. He is currently a Full Professor with the School of Electronic and Information Engineering, SCUT. His current research focuses on applied mathematics and its interdisciplinary application, including statistics learning, image and speech processing, computational intelligence, machine learning, fitting, and approximation and their applications to communication, and industrial intelligence.