Efficient Sampling for Doubly Stochastic Variational Inference in Deep Gaussian Processes Regression

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep Gaussian Processes (DGPs) enhance Gaussian Processes (GPs) in function approximation through multi-layer stacking. However, the inference of DGPs presents challenges as it has no closed-form solution. Existing methods approximate the posterior of DGPs through independent sampling and variational inference. These approaches overlook the samples' correlations and face substantial computational overhead as layers increase, hindering performance improvements. We present Efficient Deep Gaussian Processes (EDGPs) that enable efficient sampling between inner layers while maintaining full covariance characteristics. Unlike existing methods that compromise accuracy for speed, EDGP achieves high efficiency without sacrificing precision. Experiments show that EDGP has comparable, or even better performance than state-of-the-art Doubly Stochastic Deep Gaussian Processes (DSDGPs) while training is almost as efficient as basic neural networks.

1 Introduction

Gaussian Processes (GPs) are versatile tools for data analysis, offering robust modeling capabilities, broad applicability, and significant research value [1, 2, 3, 4]. A GP is primarily defined by its kernel functions, through which prior knowledge can be embedded via kernel design to enhance model performance. For instance, kernel functions can encode structural information such as periodic patterns [5], change-points in time series [6], or simulator priors for robotics [7], enabling GPs to make effective use of domain knowledge. However, the expressive power of single-layer GPs is constrained by the kernel function's accuracy in capturing data correlations. Traditional approaches often rely on handcrafted composite kernels, which require extensive design and optimization while offering limited general utility across tasks [8, 9]. An alternative paradigm seeks to parameterize kernel representations within Reproducing Kernel Hilbert Spaces (RKHS), or to use neural networks as kernel functions [10, 11]. Although these data-driven kernel learning methods aim to automate feature extraction, they incur additional computational costs during inference, and increase the risk of overfitting. Addressing these challenges demands careful optimization strategies, architectural refinements, or advanced regularization techniques, requiring a delicate balance between expressiveness and practical efficiency [12, 13].

Deep Gaussian Processes (DGPs) are a multi-layer generalization of GPs that overcome the expressive limitations while maintaining the advantages [14]. A GP can be viewed as a single-layer neural network with an infinite number of hidden units, and the way DGPs enhance GPs' performance through nested kernel modeling between layers is analogous to how deep neural networks improve performance via stacked nonlinear feature extraction [4, 15]. Furthermore, DGPs refine the covariance characteristics of the input at each inner layer, enabling a more accurate representation and automatically learning to construct an optimal kernel tailored to the data at hand.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Training DGPs presents significant challenges due to the absence of a closed-form solution for their posterior distribution [16, 17]. Early attempts to address this relied on mean-field variational approaches, which impose strong independence and Gaussianity assumptions across layers [15, 16, 17]. These restrictive assumptions severely underestimate the correlations of the posterior between layers, limiting the model's ability to capture complex hierarchical dependencies [12]. Doubly stochastic methods have emerged as a practical alternative, leveraging numerical approximations to estimate the true posterior and log-likelihood during training [12, 18, 19, 20]. Doubly Stochastic Deep Gaussian Processes (DSDGPs) employ diagonal approximations during inner-layer sampling to reduce computational complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(N)$. This trade-off sacrifices numerical precision for efficiency, and the computational overhead remains substantial, growing markedly with number of stacked layer increases. There are also approaches that modify the DGP prior and perform inference within a parametric model; these methods introduce additional approximations to ensure tractable inference [21, 22]. The spectral-based DGP methods are closely related to ours [23, 22, 24, 25], but we do not focus on posterior approximation via spectral properties, as the spectral methods are limited to stationary conditions [26, 27, 28]. A known pathology in DGPs using zero mean functions for inner layers has been reported in Duvenaud et al. [29]. Therefore, all methods used in this paper employ a linear mean function.

In this paper, we present Efficient Deep Gaussian Processes (EDGPs) that eliminate the need for compromising between efficiency and precision during inner-layer sampling. In common with many state-of-the-art GPs' approximation schemes, we start by constructing single-layer variational GPs using the Variational Free Energy (VFE) [30] approximation method, which ensures computational tractability within each layer [31]. We obtain a DGP architecture by stacking multiple such VFE-based GPs hierarchically, where the output of one layer serves as the input to the next. At this point, the posterior distributions of all but the first layer become intractable due to the integrals over the kernel's input. EDGPs overcome this hurdle by approximating the true marginal posterior through sampling from tractable conditional (on input locations) posteriors, enabling efficient inference and training. EDGPs adopt a weight-space perspective that evaluates basis functions to represent the prior distributions rather than sampling directly like other doubly stochastic methods [5, 32]. These priors will be updated to approximate the posterior distributions according to the observations (variational distributions in VFE case), thereby completing the inference propagation. This design ensures that when input locations change, which is a common scenario in most layers, only function updating is required, eliminating the need for resampling, as illustrated in Figure 1. By avoiding recomputation of inner-layer posterior means and covariances, this approach achieves a significant reduction in computational overhead. Moreover, since EDGPs avoid diagonal approximations to reduce sampling complexity, they preserve both the full covariance structure of samples and the posterior distribution correlation across layers, thereby improving modeling accuracy and theoretical rigor.

2 Background

2.1 Single-layer Gaussian Processes

A GP involves inferring a stochastic function $f : \mathbb{R}^d \to \mathbb{R}$ based on a set of N observations $\mathbf{y} = (y_1, \ldots, y_N)^\top$ at designed locations $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^\top$. We use $\mathbf{f} = f(\mathbf{X})$ as the latent function values of the observations $\mathbf{y} = \mathbf{f} + \boldsymbol{\eta}$, $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$. The prior is defined by the mean and kernel $p(\mathbf{f}; \mathbf{X}) \sim \mathcal{N}\big(m(\mathbf{X}), k(\mathbf{X}, \mathbf{X})\big)$. The likelihood $p(\mathbf{y}|\mathbf{f})$ and the prior $p(\mathbf{f}; \mathbf{X})$ have linked the observations, the input coordinates, and the random variable $\mathbf{f}$ together, allowing for the inference of the posterior. Note that a semicolon is used to distinguish between coordinate and non-coordinate random variables. To circumvent the $\mathcal{O}(N^3)$ matrix inversion in GP inference, a series of inducing points are introduced as anchor points to reduce the computational overhead. These inducing points essentially transform the GP from the original "input $\to$ output" mapping into a two-step process: "input $\to$ inducing points $\to$ output," thereby shifting the bottleneck to the size of the inducing sets M. VFE provides an expressive and robust sparse GP method and forms the foundation of the state-of-the-art research. We use the notation consistent with Salimbeni et al. [12], where $\mathbf{u} = f(\mathbf{Z})$ represents the function values at the M inducing locations $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_M)$. By the definition of a GP, the covariance features are described by the kernel function at each pair of inputs, $k(\mathbf{x}_i, \mathbf{z}_j)$. The joint probability distribution is,

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\mathbf{u}; \mathbf{Z}, \mathbf{X}) p(\mathbf{u}; \mathbf{Z}), \tag{1}$$

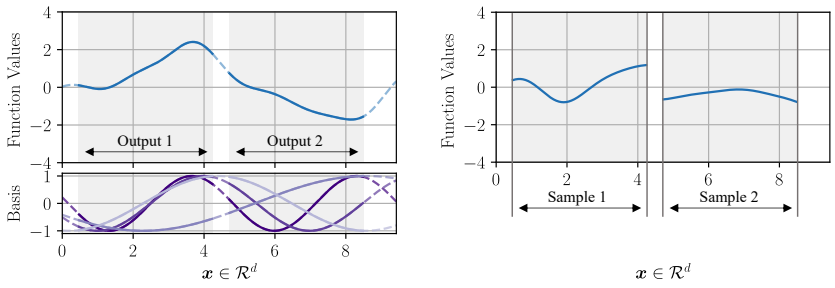

Figure 1: Illustration of two sampling approaches from a Gaussian distribution $\mathcal{N}(m(x), k(x, x))$. (a) Sampling via a weighted sum of basis functions, where the stochasticity comes from the weights and basis functions; when the input shifts, outputs at new locations can be obtained simply by re-evaluating the basis functions. (b) Direct sampling from the distribution, requires recomputing the Cholesky decomposition of the updated covariance to maintain the stochastic behavior when the input shifts.

where the prior $p(\mathbf{u}; \mathbf{Z})$ is defined as a Gaussian distribution with mean $m(\mathbf{Z})$ and covariance $k(\mathbf{Z}, \mathbf{Z})$. The conditional $p(\mathbf{f}|\mathbf{u}; \mathbf{Z}, \mathbf{X}) = \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ can be computed as a posterior using the priors $p(\mathbf{f}; \mathbf{X})$ and $p(\mathbf{u}; \mathbf{Z})$,

$$\begin{aligned} \boldsymbol{\mu} &= m(\mathbf{X}) + k(\mathbf{X}, \mathbf{Z})k(\mathbf{Z}, \mathbf{Z})^{-1}\big(\mathbf{u} - m(\mathbf{Z})\big), \\ \boldsymbol{\Sigma} &= k(\mathbf{X}, \mathbf{X}) - k(\mathbf{X}, \mathbf{Z})k(\mathbf{Z}, \mathbf{Z})^{-1}k(\mathbf{Z}, \mathbf{X}). \end{aligned} \tag{2}$$

VFE addresses sparse GPs using a variational technique. The joint probability distribution of $\mathbf{y}$, $\mathbf{f}$, and $\mathbf{u}$ is converted into the Evidence Lower Bound (ELBO) of the marginal log-likelihood objective by minimizing the Kullback-Leibler (KL) divergence between the variational posterior q and the true posterior p. Define $q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u}; \mathbf{Z}, \mathbf{X})q(\mathbf{u})$ as the factorized posterior approximation of $p(\mathbf{f}, \mathbf{u}|\mathbf{y})$, and $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{m}, \mathbf{S})$ as the approximation of $p(\mathbf{u}|\mathbf{y})$. The VFE inference solution at location $\mathbf{X}$ is given by,

$$q(\mathbf{f}; \mathbf{Z}, \mathbf{X}) = \int p(\mathbf{f}|\mathbf{u}; \mathbf{Z}, \mathbf{X})q(\mathbf{u})\mathrm{d}\mathbf{u} = \mathcal{N}(\mathbf{f}|\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}), \tag{3}$$

where the mean and covariance are,

$$\begin{aligned} \tilde{\boldsymbol{\mu}} &= m(\mathbf{X}) + k(\mathbf{X}, \mathbf{Z})k(\mathbf{Z}, \mathbf{Z})^{-1}\big(\mathbf{m} - m(\mathbf{Z})\big), \\ \tilde{\boldsymbol{\Sigma}} &= k(\mathbf{X}, \mathbf{X}) - k(\mathbf{X}, \mathbf{Z})k(\mathbf{Z}, \mathbf{Z})^{-1}\big[k(\mathbf{Z}, \mathbf{Z}) - \mathbf{S}\big]k(\mathbf{Z}, \mathbf{Z})^{-1}k(\mathbf{Z}, \mathbf{X}). \end{aligned} \tag{4}$$

The corresponding ELBO can be obtained through simple transformation [30],

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{f}; \mathbf{Z}, \mathbf{X})}\big[\log p(\mathbf{y}|\mathbf{f})\big] - \mathrm{KL}\big[q(\mathbf{u})||p(\mathbf{u}; \mathbf{Z})\big]. \tag{5}$$

The optimization in Equation 5 and the inference in Equation 4 jointly constitute the VFE workflow.

2.2 Doubly Stochastic Deep Gaussian Processes

DGPs extend the single-layer VFE by using the output of one GP layer as the input coordinates for the next, enabling the modeling of complex nonlinear features. Since the inputs in DGPs are not fixed locations but rather random variables drawn from the previous GP layer's output, the inference in Equation 3 involves an integral over the kernel function's input, thereby rendering the problem intractable,

$$q(\mathbf{f}^2; \mathbf{Z}^2, \mathbf{Z}^1, \mathbf{f}^0) = \int p(\mathbf{f}^2|\mathbf{u}^2; \mathbf{Z}^2, \mathbf{f}^1)q(\mathbf{u}^2)p(\mathbf{f}^1|\mathbf{u}^1; \mathbf{Z}^1, \mathbf{f}^0)q(\mathbf{u}^1)\mathrm{d}\mathbf{u}^2\mathrm{d}\mathbf{u}^1\mathrm{d}\mathbf{f}^1, \tag{6}$$

where we present a two-layer example with $\mathbf{f}^0$ being the input location $\mathbf{X}$.

The original DGP's formulation trivially follows the VFE structure, introducing variational techniques not only in the inducing variables but also in the noisy corruptions of the output $\mathbf{y}^l$ at each GP layer. This parameterization helps avoid the intractable integrals in the ELBO, providing a closed-form training solution. However, this design forces the inputs to each layer to be independent of the outputs from the previous layer. The variational noisy corruptions are determined separately during training, and such overly factorized DGPs essentially degenerate into single-layer GPs with independent inputs.

DSDGPs link the output of each GP layer to the input of the next. This method ensures the transfer of input information across layers, but it also makes the model intractable. A L-layer DSDGP approximates the true ELBO and inference by sampling an unbiased estimate $\hat{\mathbf{f}}^L$ of the posterior, i.e., to transform from integrating Equation 7,

$$q(\mathbf{f}^L; \mathbf{Z}^L, \ldots, \mathbf{Z}^1, \mathbf{f}^0) = \int \prod_{l=1}^{L} q(\mathbf{f}^l; \mathbf{Z}^l, \mathbf{f}^{l-1}) \mathrm{d}\mathbf{f}^{l-1}, \tag{7}$$

to recursively performing Equation 8,

$$\hat{\mathbf{f}}^l = \text{DiagSample}[q(\mathbf{f}^l; \mathbf{Z}^l, \hat{\mathbf{f}}^{l-1})], \tag{8}$$

where DiagSample conduct independently sample from a Gaussian $\mathcal{N}(a, \mathbf{A})$ as $a + \epsilon \odot \sqrt{\text{diag}(\mathbf{A})}$, $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$, and $q(\mathbf{f}^l; \mathbf{Z}^l, \hat{\mathbf{f}}^{l-1})$ can be tractably solved within each layer as Equation 3.

DSDGP avoids the cubic computational cost of Cholesky decomposition of covariance by employing a diagonal approximation when sampling from each layer's GP output distribution, and thus does not effectively utilize the covariance to model complex correlation characteristics. From this perspective, DSDGP can be seen as a diagonal, noisy-corrupted deep orthogonal projection network [33].

The core idea behind the DGP framework lies in exploring the nesting property. The output of the preceding GP will be adjusted by its second-order moment and then serve as the input to the kernel function of the following GP, thereby having a recursive influence on the output. This fundamental objective has yet to be realized in existing DGP methods. The EDGP proposed in this paper addresses this gap. By replacing the resampling step DiagSample in each layer of DSDGP with a re-evaluation, EDGP has achieved a significant reduction in computational cost while allowing a full approximation of the nested kernel.

2.3 Weight Space view of Gaussian Processes

The aforementioned methods treat $\mathbf{f}$ as a function value whose stochasticity is governed by the distributional hyperparameters. An alternative perspective is to view $\mathbf{f}$ in the weight space as a weighted sum of basis functions. The connection between these two perspectives lies in the interpretation of the kernel function $k(\cdot, \cdot)$ as the inner product between evaluation functions in an RKHS.

Random Fourier Features (RFFs) are widely adopted in training large-scale kernel machines. It serves as a basis functions that accelerate computation by mapping input data into a random low-dimensional feature space. The RFF representation in the weight-space GP is $\phi_i(\mathbf{X}) = \sqrt{2/b}\cos(\boldsymbol{\theta}_i \mathbf{X}^\top + \tau_i)$, where $\boldsymbol{\theta}_i$ are sampled from $\mathcal{N}(0, \boldsymbol{I})$ and τ_i are sampled from $U(0, 2\pi)$.

We impose a GP prior on $\mathbf{f}$ corresponding to a standard RBF kernel by defining the following Bayesian linear model,

$$\mathbf{f} = \sum_{i=1}^{b} w_i \phi_i^\top(\mathbf{X}) \qquad w_i \sim \mathcal{N}(0, 1). \tag{9}$$

Notably, in this formulation, the stochasticity of $\mathbf{f}$ is determined directly by the weights w and ϕ, rather than indirectly through the location $\mathbf{X}$ affecting the kernel matrix, as in the function-space view. The weight-space and function-space views of Gaussian processes are equivalent, and both the sparse approximation techniques and the hierarchical structures discussed earlier can be reinterpreted under the weight-space framework. However, RFF-based weighted sums cannot faithfully recover the true posterior, as the true posterior covariance is often non-stationary, while RFFs can only capture

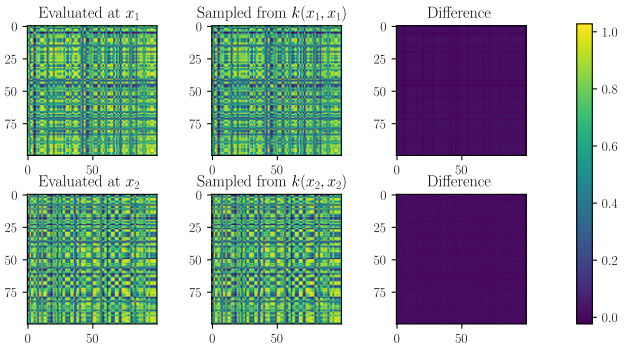

Figure 2: Validation of the effectiveness of the weight-space sampling method. The method is evaluated by comparing the difference between the sample covariance matrix obtained using basis functions at different inputs x_1 and x_2 and the covariance computed directly from the standard RBF kernel. The number of samples is 20000, and the number of basis functions is 2048.

stationary properties. This limitation has hindered the broader application of RFFs in deep Gaussian processes.

EDGP not only leverages the computational efficiency of RFFs but also overcomes their inability to model non-stationary posteriors. By successfully incorporating RFFs into a nested structure, EDGP achieves a win-win outcome of reducing computational complexity while also enhancing model performance.

3 Efficient Deep Gaussian Processes

EDGP adopts the VFE structure and features two key characteristics: first, it maintains the exact model by preserving the conditional distribution within each layer; second, it assumes that the variational distribution $q(\mathbf{u}^l)$ at each layer is a Gaussian parameterized by a mean $\mathbf{m}^l$ and covariance $\boldsymbol{S}^l$. Therefore, the joint posterior can be written in the following factorized form:

$$q(\{\mathbf{f}^l, \mathbf{u}^l\}_{l=1}^L) = \prod_{l=1}^{L} p(\mathbf{f}^l|\mathbf{u}^l; \mathbf{Z}^l, \mathbf{f}^{l-1}) q(\mathbf{u}^l). \tag{10}$$

Note that aside from replacing the fixed input with random variables, EDGP retains the VFE structure within each layer. Thus, following Equation 3, the inducing variables in each layer can still be marginalized analytically. Say that $q(\mathbf{f}^l; \mathbf{Z}^l, \mathbf{f}^{l-1}) = \int p(\mathbf{f}^l|\mathbf{u}^l; \mathbf{Z}^l, \mathbf{f}^{l-1}) q(\mathbf{u}^l) \mathrm{d}\mathbf{u}^l = \mathcal{N}(\mathbf{f}^l|\tilde{\boldsymbol{\mu}}^l, \tilde{\boldsymbol{\Sigma}}^l)$ we have,

$$\begin{aligned} \tilde{\boldsymbol{\mu}}^l &= m(\mathbf{f}^{l-1}) + k(\mathbf{f}^{l-1}, \mathbf{Z}^l) k(\mathbf{Z}^l, \mathbf{Z}^l)^{-1} \big(\mathbf{m}^l - m(\mathbf{Z}^l)\big), \\ \tilde{\boldsymbol{\Sigma}}^l &= k(\mathbf{f}^{l-1}, \mathbf{f}^{l-1}) - k(\mathbf{f}^{l-1}, \mathbf{Z}^l) k(\mathbf{Z}^l, \mathbf{Z}^l)^{-1} \big[k(\mathbf{Z}^l, \mathbf{Z}^l) - \mathbf{S}^l\big] k(\mathbf{Z}^l, \mathbf{Z}^l)^{-1} k(\mathbf{Z}^l, \mathbf{f}^{l-1}). \end{aligned} \tag{11}$$

EDGP approximates the marginal posterior distribution via sampling, with its core mechanism being a recursive sample across layers. Specifically, to approximate the marginal posterior at the l-th layer, one must first obtain samples from the posterior of the preceding layer $\hat{\mathbf{f}}^{l-1}$, as Equation 11 suggests. This sampling-based approximation presents two main challenges. First, even when the distributional form is clear, sampling incurs a time cost of $\mathcal{O}(N^3)$. Second, whenever the output of the previous GP layer changes due to updates, the subsequent GP layer resamples accordingly, increasing the computational overhead.

Proposition 1 *Let $\hat{\mathbf{f}}_q^l$, $\hat{\mathbf{f}}_p^l$, $\hat{\mathbf{u}}_q^l$ and $\hat{\mathbf{u}}_p^l$ denote samples respectively drawn from the marginal posterior $q(\mathbf{f}^l; \mathbf{Z}^l, \mathbf{f}^{l-1})$, prior $p(\mathbf{f}^l; \mathbf{f}^{l-1})$, variational distribution $q(\mathbf{u}^l)$, and prior $p(\mathbf{u}^l; \mathbf{Z}^l)$. Then $\hat{\mathbf{f}}_q^l$ can be substituted with $\tilde{\mathbf{f}}^l$ defined as follows:*

$$\tilde{\mathbf{f}}^l \overset{def}{=} \hat{\mathbf{f}}_p^l + k(\mathbf{f}^{l-1}, \mathbf{Z}^l) k(\mathbf{Z}^l, \mathbf{Z}^l)^{-1} (\hat{\mathbf{u}}_q^l - \hat{\mathbf{u}}_p^l). \tag{12}$$

Proof 1 *Proof is provided in Appendix A.*

Proposition 1 offers a novel perspective on the inference propagation: rather than sampling directly from the distribution, one can sample from the prior and apply a correction based on observations. This approach shifts the focus from studying the non-stationary posterior to sampling with a stationary prior, where weight-space methods can be employed for efficient learning.

Proposition 2 *Let $\hat{\mathbf{f}}_p^l$ be the sample drawn from the prior $p(\mathbf{f}^l; \mathbf{f}^{l-1}) = \mathcal{N}\big(m(\mathbf{f}^{l-1}), k(\mathbf{f}^{l-1}, \mathbf{f}^{l-1})\big)$. Then $\hat{\mathbf{f}}_p^l$ can be substituted with the following expression:*

$$\sum_{i=1}^{b} w_i \phi_i^\top(\mathbf{f}^{l-1}) + m(\mathbf{f}^{l-1}). \tag{13}$$

Proof 2 *Proof is provided in Appendix B.*

Sample from the marginal posterior By incorporating Proposition 2 and Proposition 1, the recursive computation of EDGP's marginal posterior distribution can thus be summarized as follows: first, use RFF to sample from both the $\mathbf{f}$ and $\mathbf{u}$ prior in the weight space; then, adjust the prior samples based on observations to approximate posterior samples; finally, feed these posterior samples as input locations into the next-layer GP to determine its prior covariance. The sample procedure is listed in Algorithm 1.

Algorithm 1 Sample from the marginal posterior

1: **Input:** input locations $\mathbf{X}$.
2: **Compute:** $\tilde{\mathbf{f}}^l$ for each layer.
3: **Initialize:** $\mathbf{f}^0$ is set to $\mathbf{X}$, initialize kernel $k(\cdot, \cdot)$.
4: **for** $l = 1, \ldots, L-1$ **do**
5: Sample $\hat{\mathbf{f}}_p^l = \sum_{i=1}^{b} w_i \phi_i^\top(\mathbf{f}^{l-1}) + m(\mathbf{f}^{l-1})$, $\hat{\mathbf{u}}_p^l = \sum_{i=1}^{b} w_i \phi_i^\top(\mathbf{Z}^l) + m(\mathbf{Z}^l)$.
6: Sample $\hat{\mathbf{u}}_q^l \sim q(\mathbf{u}^l)$.
7: Compute: $\tilde{\mathbf{f}}^l = \hat{\mathbf{f}}_p^l + k(\mathbf{f}^{l-1}, \mathbf{Z}^l) k(\mathbf{Z}^l, \mathbf{Z}^l)^{-1} (\hat{\mathbf{u}}_q^l - \hat{\mathbf{u}}_p^l)$.
8: Set: $\mathbf{f}^l = \tilde{\mathbf{f}}^l$.
9: **end for**

Computation of the ELBO We compute the objective of EDGP in the same manner as VFE; the ELBO can be obtained through Jensen's inequality on the marginal log-likelihood,

$$\mathcal{L} = \mathbb{E}_{q(\{\mathbf{f}^l, \mathbf{u}^l\}_{l=1}^L)} \log \left[\frac{p(\mathbf{y}|\mathbf{f}^L) \prod_{l=1}^{L} p(\mathbf{f}^l|\mathbf{u}^l; \mathbf{Z}^l, \mathbf{f}^{l-1}) p(\mathbf{u}^l)}{q(\{\mathbf{f}^l, \mathbf{u}^l\}_{l=1}^L)} \right]. \tag{14}$$

After simplifying and consolidating terms, the final expression for Equation 14 is obtained:

$$\mathcal{L} = \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{f}_i^L; \mathbf{Z}^L, \tilde{\mathbf{f}}^{L-1})} [\log p(\mathbf{y}_i|\mathbf{f}_i^L)] - \sum_{l=1}^{L} \mathrm{KL}[q(\mathbf{u}^l)||p(\mathbf{u}^l; \mathbf{Z}^l)], \tag{15}$$

where subscript i denotes the ith component.

Comparison with DSDGP Although EDGP and DSDGP share the same theoretical computational complexity due to their common variational inference framework, EDGP demonstrates significantly faster empirical performance. This efficiency stems from EDGP's compact computational structure, where with one-step computation (Equation 12) it captures both the posterior mean and covariance during sampling. In contrast, DSDGP requires explicit computation of the bias term and covariance, incurring substantial additional overhead that slows down computation.

More importantly, DSDGP achieves the same theoretical computational complexity as EDGP only under a diagonal approximation. If DSDGP attempts to restore full covariance during posterior sampling, its complexity escalates to $\mathcal{O}(N^3)$. In comparison, EDGP constructs an efficient DGP that retains the full covariance characteristics without compromising on structural assumptions or predictive performance, addressing a long-standing challenge in this field.

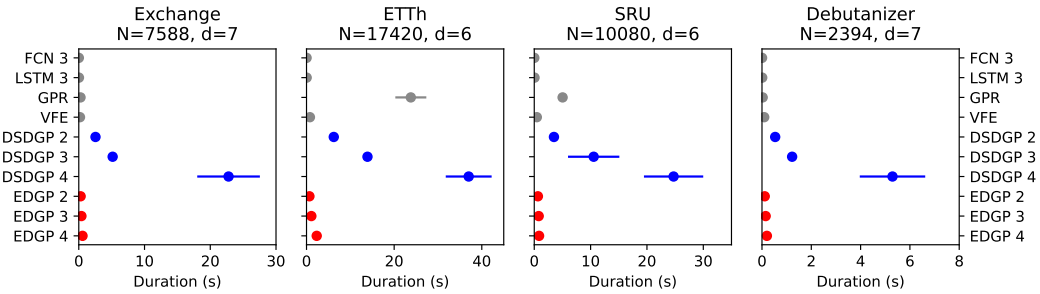

Figure 3: Runtime comparison of all methods across the four datasets.

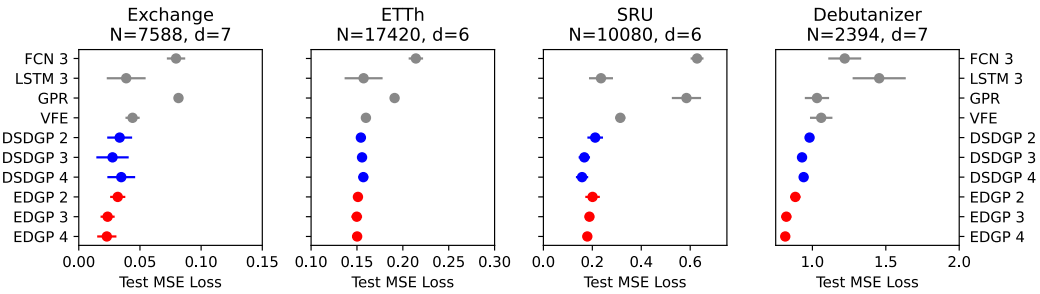

Figure 4: Performance comparison of all methods across the four datasets on MSE metric. GPR and VFE are aligned for comparison using a linear mapping $m(\mathbf{X}) = \mathbf{X}W$ as their prior mean.

4 Experiments and Analysis

4.1 Experiments Setup

We evaluate EDGP on four mainstream regression benchmark datasets. The **ETTh** [34] dataset consists of hourly load and oil temperature data from electricity transformers collected between July 2016 and July 2018. The **Exchange** [35] dataset records daily exchange rates for eight countries from 1990 to 2016. The **SRU** [36] dataset captures residual SO_2 concentrations in tail gas emissions during the oxidative removal of H_2S at a large industrial refinery. The **Debutanizer** [36] dataset contains butane concentration measurements from a debutanizer column in naphtha separation units within petroleum production. These datasets span common real-world regression scenarios and vary in modeling difficulty: ETT and Debutanizer are more challenging with lower reported accuracies, while Exchange and SRU are relatively easier and have higher existing fit precision.

We aim to compare EDGP's performance and speed against DSDGP and classic GP models, including traditional full GP and variational sparse GP. We aim to show how EDGP achieves both faster computation and higher predictive accuracy. To strengthen the comparison, we also include two well-established neural regression models, Long Short-Term Memory (LSTM) [37] and Fully Connected Network (FCN).

We record detailed results for EDGP and DSDGP with GP layer depths set to 2, 3, and 4. For the neural network baselines, we use 3 layers, striking a balance between avoiding overfitting and retaining sufficient feature extraction capacity. All other experimental hyperparameters are held constant across models. Inputs are preprocessed as a moving-average model of order 16 [38], which corresponds to a sequence length of 16 for LSTM models. Hidden dimensions across all layers are fixed at 64, and the RBF kernel is used uniformly for all GP layers and models. Both EDGP and DSDGP propagate 20 samples at each inner layer. The best validation performance is recorded on the last 800 data points for all methods and datasets. The number of inducing points is set to 256 for all datasets. The number of basis function is set to 2048 for EDGP. All experiments are conducted on a workstation with an AMD R7-5800 CPU and an NVIDIA RTX 3060 GPU.

Table 1: Regression MSE and MAE results

Datasets		Exchange		ETTh		SRU		Debutanizer	
Models	Layers	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
FCN	3	0.0795	0.2261	0.2142	0.3694	0.6280	0.5817	1.2202	0.8486
LSTM	3	0.0388	0.1600	0.1572	0.3013	0.2354	0.3517	1.4549	0.8641
GPR	N/A†	0.0815	0.2603	0.1910	0.3499	0.5847	0.5192	1.0311	0.7808
VFE	N/A†	0.0440	0.1666	0.1598	0.3141	0.3143	0.4012	1.0600	0.8005
DSDGP	2	0.0334	0.1469	0.1543	0.3103	0.2117	0.3588	0.9807	0.7786
DSDGP	3	0.0276	0.1233	0.1555	0.3119	0.1673	0.3176	0.9294	0.7542
DSDGP	4	0.0347	0.1364	0.1569	0.3135	0.1580	0.3133	0.9404	0.7599
EDGP	**2**	**0.0318**	**0.1432**	**0.1511**	**0.3079**	**0.2009**	**0.3479**	**0.8837**	**0.7289**
EDGP	**3**	**0.0236**	**0.1193**	**0.1498**	**0.3086**	**0.1882**	**0.3391**	**0.8225**	**0.6952**
EDGP	**4**	**0.0229**	**0.1151**	**0.1502**	**0.3100**	**0.1795**	**0.3360**	**0.8151**	**0.6907**

† N/A stands for Not Accessible, meaning such methods have no attribute of stacked layers. VFE can be viewed as a 1-layer DSDGP/EDGP.

4.2 Result Analysis

We first present a comparison of the runtime efficiency of EDGP with that of other baseline methods. To this end, we record the duration required for each model to train an epoch over the dataset and report the mean and standard deviation across 20 runs in Figure 3.

Both DSDGP and EDGP employ an unbiased mini-batch training technique to achieve scalability. Despite their $\mathcal{O}(N)$ computational complexity making batch size theoretically irrelevant to the comparative results, we still choose a relatively large batch size. Note that the VFE method is not originally proposed as an observation-factorized approach (Equation 16 in [30]). However, for a fair comparison with EDGP and DSDGP, we apply the same sub-sampling strategy to convert VFE into a factorized parametric method (Equation 13 in [30]). Since VFE also has $\mathcal{O}(N)$ complexity, this adjustment does not affect the validity of the comparison. All models (LSTM, FCN, EDGP, DSDGP, and VFE) are trained with a batch size of 1024, while GPR is updated using the entire dataset.

Experiments show that GPR requires significantly more training time than VFE and EDGP, especially on large datasets, which is a reasonable outcome given GPR's cubic computational complexity. What stands out is that DSDGP, despite being a $\mathcal{O}(N)$ method, exhibits a runtime comparable (or even higher) to GPR across all datasets. Even on smaller datasets, Debutanizer, the 4-layer DSDGP incurs almost 20 times higher training overhead compared to other methods. Despite using diagonal approximation techniques to reduce the computational burden, DSDGP's runtime increases sharply with depth. Across all datasets, the jump in training time from 3 to 4 layers is particularly steep, suggesting that very deep DSDGP models may not be practically usable. In contrast, EDGP's training durations maintain stable behaviour: not only is its computational cost moderate, but the additional overhead from increasing the number of layers appears to grow linearly.

Furthermore, DSDGP's significant computational cost does not translate into equivalent better performance. Figure 4 shows the mean and standard deviation of MSE loss over 20 independent trials for each method on every dataset.

Note that for DSDGP, the VFE can be viewed as its single-layer variant. While stacking more layers generally improves performance, the gains are relatively modest compared to the significant increase in training time. This suggests that DSDGP is not well-suited for deep architectures.

In contrast, EDGP demonstrates a clear advantage in constructing deep frameworks. As shown in Figure 4, EDGP consistently outperforms DSDGP in most scenarios and benefits more noticeably from deeper architectures, without showing signs of overfitting as DSDGP does. Meanwhile, EDGP also requires substantially less training time than DSDGP, making it more practical in real-world applications.

We would like to highlight why GPR and VFE exhibit stochasticity in Figure 4. Note that DSDGP and EDGP do not adopt the traditional zero-mean prior; therefore, GPR and VFE are aligned

for comparison using a linear mapping $m(\mathbf{X}) = \mathbf{X}W$ as their prior mean. This linear mapping is randomly initialized following the Kaiming initialization method [39], introducing stochasticity into the models. Additionally, since VFE's ELBO is obtained by log-likelihood minus KL divergence, this also contributes to its stochasticity.

Beyond the visual comparisons in Figures 3 and 4, Table 1 presents the quantitative performance of all models. It is worth noting that EDGP tends to achieve its best performance at a depth of 4 layers, while the performance of 4-layer DSDGP models is often worse than that of their shallow counterparts. This further supports the claim that EDGP is better suited for deep architectures. When comparing the best performance of EDGP with the best results from competing methods, we observe substantial improvements. For example, on the Exchange dataset, the best EDGP MSE is 0.0229 with 4 layers, representing a 17.03% improvement over the second-best DSDGP (3 layers) with an MSE of 0.0276. On the ETTh dataset, the best EDGP result is 0.1498 (3 layers), improving upon the second-best DSDGP (2 layers) at 0.1543 by 2.92%. On the SRU dataset, EDGP with 4 layers achieves an MSE of 0.1795, which is slightly worse than DSDGP's 0.1580 with the same depth. On the Debutanizer dataset, EDGP (4 layers) reaches an MSE of 0.8151, significantly outperforming the second-best DSDGP (3 layers) at 0.9294, by 12.30%.

As for why EDGP underperforms DSDGP on the SRU dataset, we provide a conjecture in Section 5. Nevertheless, the overall results strongly validate the effectiveness of EDGP and highlight its contribution to advancing Gaussian process research.

5 Discussion and Limitation

Experiments demonstrate that EDGP is effective and performs well across a range of datasets. While DSDGP gains only modest benefits from additional layers due to increased computational costs, EDGP shows clear and significant advantages as the depth increases.

We would like to discuss why EDGP does not vastly outperform DSDGP in all scenarios and offer a conjecture. The essence of GPs lies in the assumption that the correlation between input locations reflects the correlation between target outputs, i.e., closer inputs yield more similar outputs. The key difference between EDGP and DSDGP lies in how the inner layers are handled: DSDGP computes the posterior mean but ignores the posterior covariance in subsequent inference, thus preserving the original input correlation characteristic. In contrast, EDGP refines this structure by incorporating the posterior covariance to adjust the inputs to the next layer. Therefore, on datasets where the correlation structure between inputs and outputs is well-aligned (i.e., easier datasets like SRU), DSDGP can match or even slightly outperform EDGP. However, on more challenging datasets with possible misaligned correlations, e.g., Debutanizer, DSDGP falls short, whereas EDGP's additional adjustment yields significantly better performance.

While EDGP demonstrates clear advantages in accuracy and efficiency, these gains come at the cost of kernel flexibility. At its core, EDGP transforms function-space sampling into weight-space sampling, where weights follow independent Gaussian distributions, allowing for efficient linear-time complexity. However, this transformation inherently limits the method to RBF kernels. While extensions to other stationary kernels are theoretically possible, the resulting weight distributions may not allow equally efficient sampling. For non-stationary kernels, EDGP is not directly applicable.

6 Conclusion

We have presented a novel DGP method termed EDGP which performs efficient and effective inference. Both theoretical and empirical analyses show that EDGP addresses the fundamental trade-off in DGPs between computational efficiency and inference accuracy. Experiment results demonstrate that EDGP significantly outperforms DSDGP in runtime while achieving equal or better predictive performance. This advantage arises from replacing inner-layer sampling with basis-function decomposition and posterior correction, thus retaining full covariance structure without additional overhead.

References

[1] Marc Peter, D., F. Dieter, R. Carl Edward. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423, 2015.

[2] Giulio, G., C. Ruggero, R. Diego, et al. A black-box physics-informed estimator based on gaussian process regression for robot inverse dynamics identification. *IEEE Transactions on Robotics*, 40:4820–4836, 2024.

[3] François-Xavier, B., O. Chris J., G. Mark, et al. Probabilistic integration: A role in statistical computation? *Statistical Science*, 34(1):pp. 1–22, 2019.

[4] Vincent, D., H. James, V. D. W. Mark, et al. Deep neural networks as point estimates for deep gaussian processes. In *Advances in Neural Information Processing Systems*, vol. 34, pages 9443–9455. Curran Associates, Inc., 2021.

[5] Carl Edward, R., W. Christopher K. I. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[6] Roman, G., O. Michael A., R. Stephen J. Sequential bayesian prediction in the presence of changepoints. In *Proceedings of the Twenty-sixth Annual International Conference on Machine Learning*, ICML '09, page 345–352. Association for Computing Machinery, New York, NY, USA, 2009.

[7] Mark, C., H. Jonathan P. Efficient reinforcement learning for robots using informative simulated priors. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2605–2612. 2015.

[8] Hugh, S. *Deep Gaussian Processes: Advances in Models and Inference*. Ph.D. thesis, Imperial College London, 2019.

[9] David, D., L. James, G. Roger, et al. Structure discovery in nonparametric regression through compositional kernel search. In *Proceedings of the Thirtieth International Conference on Machine Learning*, vol. 28 of *Proceedings of Machine Learning Research*, pages 1166–1174. PMLR, Atlanta, Georgia, USA, 2013.

[10] Ching-An, C., B. Byron. Incremental variational sparse gaussian process regression. In *Advances in Neural Information Processing Systems*, vol. 29. Curran Associates, Inc., 2016.

[11] Calandra, R., J. Peters, C. E. Rasmussen, et al. Manifold gaussian processes for regression. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3338–3345. 2016.

[12] Hugh, S., D. Marc. Doubly stochastic variational inference for deep gaussian processes. In *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[13] Wilson, A. G., P. Izmailov, M. D. Hoffman, et al. Evaluating approximate inference in bayesian deep learning. In *Proceedings of the NeurIPS 2021 Competitions and Demonstrations Track*, vol. 176 of *Proceedings of Machine Learning Research*, pages 113–124. PMLR, 2022.

[14] Matthew M., D., G. Mark A., S. Andrew M., et al. How deep are deep gaussian processes? *Journal of Machine Learning Research*, 19(54):1–46, 2018.

[15] Andreas, D., L. Neil D. Deep gaussian processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, vol. 31 of *Proceedings of Machine Learning Research*, pages 207–215. PMLR, 2013.

[16] Zhenwen, D., D. Andreas, G. Javier, et al. Variational auto-encoded deep gaussian processes. In *Proceedings of the International Conference on Learning Representations*, vol. 3. Caribe Hotel, San Juan, PR, 2016.

[17] César Lincoln C., M., D. Zhenwen, D. Andreas, et al. Recurrent Gaussian processes. In *Proceedings of the International Conference on Learning Representations*, vol. 3. Caribe Hotel, San Juan, PR, 2016.

[18] Boris, L., C. Ronald R., K. Yuval. Doubly stochastic normalization of the gaussian kernel is robust to heteroskedastic noise. *SIAM Journal on Mathematics of Data Science*, 3(1):388–413, 2021.

[19] Adam, V., S. Eleftheriadis, A. Artemev, et al. Doubly sparse variational gaussian processes. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and*

Statistics, vol. 108 of *Proceedings of Machine Learning Research*, pages 2874–2884. PMLR, 2020.

[20] Naiqi, L., L. Wenjie, S. Jifeng, et al. Stochastic deep gaussian processes over graphs. In *Advances in Neural Information Processing Systems*, vol. 33, pages 5875–5886. Curran Associates, Inc., 2020.

[21] Thang, B., H.-L. Daniel, H.-L. Jose, et al. Deep gaussian processes for regression using approximate expectation propagation. In *Proceedings of The Thirty-third International Conference on Machine Learning*, vol. 48 of *Proceedings of Machine Learning Research*, pages 1472–1481. PMLR, New York, New York, USA, 2016.

[22] Kurt, C., B. Edwin V., M. Pietro, et al. Random feature expansions for deep Gaussian processes. In *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, pages 884–893. PMLR, 2017.

[23] Mojmir, M., K. Andreas. Efficient high dimensional bayesian optimization with additivity and quadrature fourier features. In *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018.

[24] Tim G. J., R., S. Dino, G. Yarin. Inter-domain deep Gaussian processes. In *Proceedings of the 37th International Conference on Machine Learning*, vol. 119 of *Proceedings of Machine Learning Research*, pages 8286–8294. PMLR, 2020.

[25] James, H., D. Nicolas, S. Arno. Variational fourier features for gaussian processes. *Journal of Machine Learning Research*, 18(151):1–52, 2018.

[26] James, W., B. Viacheslav, T. Alexander, et al. Efficiently sampling functions from gaussian process posteriors. In *Proceedings of the 37th International Conference on Machine Learning*, vol. 119 of *Proceedings of Machine Learning Research*, pages 10292–10302. PMLR, 2020.

[27] Carl Edward, R., Q.-C. Joaquin. Healing the relevance vector machine through augmentation. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML, page 689–696. Association for Computing Machinery, New York, NY, USA, 2005.

[28] Daniele, C., C. Luigi, L. Alessandro, et al. Gaussian process optimization with adaptive sketching: Scalable and no regret. In *Proceedings of the Thirty-Second Conference on Learning Theory*, vol. 99 of *Proceedings of Machine Learning Research*, pages 533–557. PMLR, 2019.

[29] David, D., R. Oren, A. Ryan, et al. Avoiding pathologies in very deep networks. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, vol. 33 of *Proceedings of Machine Learning Research*, pages 202–210. PMLR, Reykjavik, Iceland, 2014.

[30] Michalis, T. Variational model selection for sparse gaussian process regression. *Report, University of Manchester, UK*, 2009.

[31] David, B., R. Carl Edward, V. D. W. Mark. Rates of convergence for sparse variational Gaussian process regression. In *Proceedings of the Thirty-sixth International Conference on Machine Learning*, vol. 97 of *Proceedings of Machine Learning Research*, pages 862–871. PMLR, 2019.

[32] Hugh, S., E. Stefanos, H. James. Natural gradients in practice: Non-conjugate variational inference in gaussian process models. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, vol. 84 of *Proceedings of Machine Learning Research*, pages 689–697. PMLR, 2018.

[33] Shaoqi, W., Y. Chunjie, L. Siwei. Approximated orthogonal projection unit: Stabilizing regression network training using natural gradient. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024.

[34] Haoyi, Z., Z. Shanghang, P. Jieqi, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, 2021.

[35] Guokun, L., C. Wei-Cheng, Y. Yiming, et al. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 95–104. Association for Computing Machinery, New York, NY, USA, 2018.

[36] Luigi, F., G. Salvatore, R. Alessandro, et al. *Soft Sensors for Monitoring and Control of Industrial Processes*, vol. 22. Springer, 2007. Springer.

[37] Maximilian, B., P. Korbinian, S. Markus, et al. xlstm: Extended long short-term memory. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024.

[38] Jiahao, S., W. Shiqi, H. Furong. Arma nets: Expanding receptive field for dense prediction. In *Advances in Neural Information Processing Systems*, vol. 33, pages 17696–17707. Curran Associates, Inc., 2020.

[39] Kaiming, H., Z. Xiangyu, R. Shaoqing, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*. 2015.

[40] Ali, R., R. Benjamin. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, vol. 20. Curran Associates, Inc., 2007.

A Proof of Proposition 1

To prove that $\hat{\mathbf{f}}_q^l$ can be substituted by $\tilde{\mathbf{f}}^l$, we only need to focus on whether these two have same mean and covariance. To facilitate the proof, we would like to pre-define the following notations,

$$\begin{aligned}\mathbb{E}_x[a] &\overset{\text{def}}{=} \int ap(x)\mathrm{d}x,\\ \mathbb{E}_y\mathbb{E}_{x|y}[a] &\overset{\text{def}}{=} \int \left(\int ap(x|y)\mathrm{d}x\right)p(y)\mathrm{d}y = \mathbb{E}_x[a],\\ \mathbb{D}_{x|y}(a) &\overset{\text{def}}{=} \mathbb{E}_{x|y}\left[(a-\mathbb{E}_{x|y}[a])(a-\mathbb{E}_{x|y}[a])^\top\right].\end{aligned} \tag{16}$$

For clarity demonstration we rewrite Equation 12 in the following,

$$\tilde{\mathbf{f}}^l \overset{\text{def}}{=} \hat{\mathbf{f}}_p^l + k(\mathbf{f}^{l-1},\mathbf{Z}^l)k(\mathbf{Z}^l,\mathbf{Z}^l)^{-1}(\hat{\mathbf{u}}_q^l - \hat{\mathbf{u}}_p^l).$$

It is straightforward to see that $\hat{\mathbf{f}}_q^l$ shares the same mean with $\tilde{\mathbf{f}}^l$. We omit the ***hat*** superscript to transform the notation from samples to random variables. $\tilde{\mathbf{f}}^l$ is also now seen as a complex random variable instead of a sample. The expectation of $\tilde{\mathbf{f}}^l$ is computed through $p(\mathbf{f}^l)$, $q(\mathbf{u}^l)$, and $p(\mathbf{u}^l)$ from which the $\tilde{\boldsymbol{\mu}}^l$ (from Equation 11) is restored, therefore is validated.

$$\mathbb{E}_{\tilde{\mathbf{f}}^l}[\tilde{\mathbf{f}}^l] = \mathbb{E}_{\mathbf{f}_p^l}[\mathbf{f}_p^l] + k(\tilde{\mathbf{f}}^{l-1},\mathbf{Z}^l)k(\mathbf{Z}^l,\mathbf{Z}^l)^{-1}(\mathbb{E}_{\mathbf{u}_q^l}[\mathbf{u}_q^l] - \mathbb{E}_{\mathbf{u}_p^l}[\mathbf{u}_p^l]). \tag{17}$$

To pave the way for the proof of $\tilde{\mathbf{f}}^l$ covariance, we need to prove the following intermediate result.

$$\mathbb{D}_x(x) = \mathbb{E}_y[\mathbb{D}_{x|y}(x)] + \mathbb{D}_y(\mathbb{E}_{x|y}[x]). \tag{18}$$

We present the proof in the following Equation 19,

$$\begin{aligned}&\mathbb{E}_x\left[(x-\mathbb{E}_x[x])(x-\mathbb{E}_x[x])^\top\right]\\ &= \mathbb{E}_y\mathbb{E}_{x|y}\left[(x-\mathbb{E}_{x|y}[x]+\mathbb{E}_{x|y}[x]-\mathbb{E}_x[x])(x-\mathbb{E}_{x|y}[x]+\mathbb{E}_{x|y}[x]-\mathbb{E}_x[x])^\top\right]\\ &= \mathbb{E}_y\mathbb{E}_{x|y}\left[(x-\mathbb{E}_{x|y}[x])(x-\mathbb{E}_{x|y}[x])^\top + (x-\mathbb{E}_{x|y}[x])(\mathbb{E}_{x|y}[x]-\mathbb{E}_x[x])^\top\right] +\\ &\quad \mathbb{E}_y\mathbb{E}_{x|y}\left[(\mathbb{E}_{x|y}[x]-\mathbb{E}_x[x])(x-\mathbb{E}_{x|y}[x])^\top + (\mathbb{E}_{x|y}[x]-\mathbb{E}_x[x])(\mathbb{E}_{x|y}[x]-\mathbb{E}_x[x])^\top\right]\\ &= \mathbb{E}_y\mathbb{E}_{x|y}\left[(x-\mathbb{E}_{x|y}[x])(x-\mathbb{E}_{x|y}[x])^\top + (\mathbb{E}_{x|y}[x]-\mathbb{E}_x[x])(\mathbb{E}_{x|y}[x]-\mathbb{E}_x[x])^\top\right]\\ &= \mathbb{E}_y[\mathbb{D}_{x|y}(x)] + \mathbb{D}_y(\mathbb{E}_{x|y}[x])\\ &= \mathbb{D}_x(x),\end{aligned} \tag{19}$$

where the first and second equality come from the formula expansion, the third equality comes from the fact that $\mathbb{E}_y\mathbb{E}_{x|y}\left[(\mathbb{E}_{x|y}[x]-\mathbb{E}_x[x])(x-\mathbb{E}_{x|y}[x])^\top\right] = 0$ as $\mathbb{E}_{x|y}[(x-\mathbb{E}_{x|y}[x])] = 0$ and $(\mathbb{E}_{x|y}[x]-\mathbb{E}_x[x])$ is independent of x, the fourth equality comes from Equation 16, and the fifth equality comes from the definition.

Through Equation 18 we can compute the covariance of $\tilde{\mathbf{f}}^l$ by the following,

$$\begin{aligned}&\mathbb{D}_{\tilde{\mathbf{f}}^l}(\tilde{\mathbf{f}}^l)\\ &= \mathbb{E}_{\mathbf{u}_q^l}[\mathbb{D}_{\tilde{\mathbf{f}}^l|\mathbf{u}_q^l}(\tilde{\mathbf{f}}^l)] + \mathbb{D}_{\mathbf{u}_q^l}(\mathbb{E}_{\tilde{\mathbf{f}}^l|\mathbf{u}_q^l}[\tilde{\mathbf{f}}^l])\\ &= \mathbb{E}_{\mathbf{u}_q^l}\left[\mathbb{E}_{\mathbf{u}_p^l}[\mathbb{D}_{\tilde{\mathbf{f}}^l|\mathbf{u}_q^l,\mathbf{u}_p^l}(\tilde{\mathbf{f}}^l)] + \mathbb{D}_{\mathbf{u}_p^l}(\mathbb{E}_{\tilde{\mathbf{f}}^l|\mathbf{u}_q^l,\mathbf{u}_p^l}[\tilde{\mathbf{f}}^l])\right] + \mathbb{D}_{\mathbf{u}_q^l}(\mathbb{E}_{\tilde{\mathbf{f}}^l|\mathbf{u}_q^l}[\tilde{\mathbf{f}}^l]).\end{aligned} \tag{20}$$

For the last term of Equation 20 $\mathbb{D}_{\mathbf{u}_q^l}(\mathbb{E}_{\tilde{\mathbf{f}}^l|\mathbf{u}_q^l}[\tilde{\mathbf{f}}^l])$ we have,

$$\begin{aligned}&\mathbb{D}_{\mathbf{u}_q^l}(\mathbb{E}_{\tilde{\mathbf{f}}^l|\mathbf{u}_q^l}[\tilde{\mathbf{f}}^l])\\ &= \mathbb{D}_{\mathbf{u}_q^l}\left(m(\mathbf{f}^{l-1}) + k(\tilde{\mathbf{f}}^{l-1},\mathbf{Z}^l)k(\mathbf{Z}^l,\mathbf{Z}^l)^{-1}(\mathbf{u}_q^l - m(\mathbf{Z}^l))\right)\\ &= k(\tilde{\mathbf{f}}^{l-1},\mathbf{Z}^l)k(\mathbf{Z}^l,\mathbf{Z}^l)^{-1}\mathbf{S}^l k(\mathbf{Z}^l,\mathbf{Z}^l)^{-1}k(\mathbf{Z}^l,\tilde{\mathbf{f}}^{l-1}).\end{aligned} \tag{21}$$

For the second term of Equation 20 $\mathbb{D}_{\mathbf{u}_p^l}(\mathbb{E}_{\tilde{\mathbf{f}}^l|\mathbf{u}_q^l,\mathbf{u}_p^l}[\tilde{\mathbf{f}}^l])$ we have,

$$\begin{aligned}
&\mathbb{D}_{\mathbf{u}_p^l}(\mathbb{E}_{\tilde{\mathbf{f}}^l|\mathbf{u}_q^l,\mathbf{u}_p^l}[\tilde{\mathbf{f}}^l]) \\
&= \mathbb{D}_{\mathbf{u}_p^l}\left(m(\tilde{\mathbf{f}}^{l-1}) + k(\tilde{\mathbf{f}}^{l-1},\mathbf{Z}^l)k(\mathbf{Z}^l,\mathbf{Z}^l)^{-1}\left(\mathbf{u}_p^l - m(\mathbf{Z}^l)\right) + k(\tilde{\mathbf{f}}^{l-1},\mathbf{Z}^l)k(\mathbf{Z}^l,\mathbf{Z}^l)^{-1}(\mathbf{u}_q^l - \mathbf{u}_p^l)\right) \\
&= \mathbb{D}_{\mathbf{u}_p^l}\left(m(\tilde{\mathbf{f}}^{l-1}) + k(\tilde{\mathbf{f}}^{l-1},\mathbf{Z}^l)k(\mathbf{Z}^l,\mathbf{Z}^l)^{-1}\left(\mathbf{u}_q^l - m(\mathbf{Z}^l)\right)\right) \\
&= 0,
\end{aligned} \tag{22}$$

where we use the mean property of $p(\mathbf{f}_p^l|\mathbf{u}_p^l,\mathbf{Z}^l,\tilde{\mathbf{f}}^{l-1})$ from Equation 2 in the first equality, and the second and third equality come from the fact that a constant has zero covariance.

For the first term of Equation 20 $\mathbb{E}_{\mathbf{u}_p^l}\left[\mathbb{D}_{\tilde{\mathbf{f}}^l|\mathbf{u}_q^l,\mathbf{u}_p^l}(\tilde{\mathbf{f}}^l)\right]$ we have,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{u}_p^l}\left[\mathbb{D}_{\tilde{\mathbf{f}}^l|\mathbf{u}_q^l,\mathbf{u}_p^l}(\tilde{\mathbf{f}}^l)\right] \\
&= \mathbb{E}_{\mathbf{u}_p^l}\left[k(\tilde{\mathbf{f}}^{l-1},\tilde{\mathbf{f}}^{l-1}) - k(\tilde{\mathbf{f}}^{l-1},\mathbf{Z}^l)k(\mathbf{Z}^l,\mathbf{Z}^l)^{-1}k(\mathbf{Z}^l,\tilde{\mathbf{f}}^{l-1})\right] \\
&= k(\tilde{\mathbf{f}}^{l-1},\tilde{\mathbf{f}}^{l-1}) - k(\tilde{\mathbf{f}}^{l-1},\mathbf{Z}^l)k(\mathbf{Z}^l,\mathbf{Z}^l)^{-1}k(\mathbf{Z}^l,\tilde{\mathbf{f}}^{l-1}),
\end{aligned} \tag{23}$$

where the first equality comes from the fact that the covariance of $p(\mathbf{f}_p^l|\mathbf{u}_p^l,\mathbf{Z}^l,\tilde{\mathbf{f}}^{l-1})$ is independent of the observation/realization of $\mathbf{u}_p^l$.

The key to the above derivation is to recognize the difference between conditioning on $\mathbf{u}_p^l$ and conditioning on $\mathbf{u}_q^l$: the former changes the distribution of $\tilde{\mathbf{f}}^l$ while the latter does not. Combining these three parts, we restore the $\tilde{\mathbf{\Sigma}}^l$ from Equation 11, therefore completing the proof.

B Proof of Proposition 2

The key to efficient sampling from the prior lies in restoring the correct covariance structure. Therefore, we would like to show that the sample covariance obtained from Equation 9 converges in probability to the target kernel $k(\cdot,\cdot)$. This paper focuses on stationary kernels and follows the approach of RFF, which uses Fourier transforms to approximate kernel behavior [40].

Bochner's theorem ensures that the Fourier transform of any positive definite, shift-invariant kernel is a non-negative measure. If the kernel is properly scaled, its Fourier transform $p(\theta)$ becomes a valid probability distribution [40]:

$$k(x,y) = k(x-y) = \int p(\theta)e^{j\theta(x-y)}\mathrm{d}\theta = \mathbb{E}_\theta\left[\zeta_\theta(x)\zeta_\theta(y)^*\right], \tag{24}$$

where $\zeta_\theta(x)$ is defined as $e^{j\theta x}$, and $\zeta_\theta(x)\zeta_\theta(y)^*$ is an unbiased estimator of $k(x,y)$ when θ is drawn from $p(\theta)$.

Since all inputs and outputs are real-valued, only the real part of $\zeta_\theta(x)$ contributes to the computation. Thus, $e^{j\theta(x-y)}$ can be simplified to $\cos\left(\theta(x-y)\right)$. To recover an inner product structure similar to $\zeta_\theta(x)\zeta_\theta(y)^*$, we introduce an additional random variable b and apply the following transformation:

$$\begin{aligned}
&2\cos\left(\theta x + b\right)\cos\left(\theta y + b\right) \\
&= \cos\left((\theta x + b) - (\theta y + b)\right) + \cos\left((\theta x + b) + (\theta y + b)\right) \\
&= \cos\left(\theta(x-y)\right) + \cos\left(\theta(x+y) + 2b\right).
\end{aligned} \tag{25}$$

This allows the kernel $k(x,y)$ to be approximated in probability by using basis functions $\sqrt{2}\cos\left(\theta x + b\right)$, provided that the term $\cos\left(\theta(x+y)+2b\right)$ can be canceled out when b is uniformaly sampled from $U(0,2\pi)$, thus effectively restoring the kernel structure.

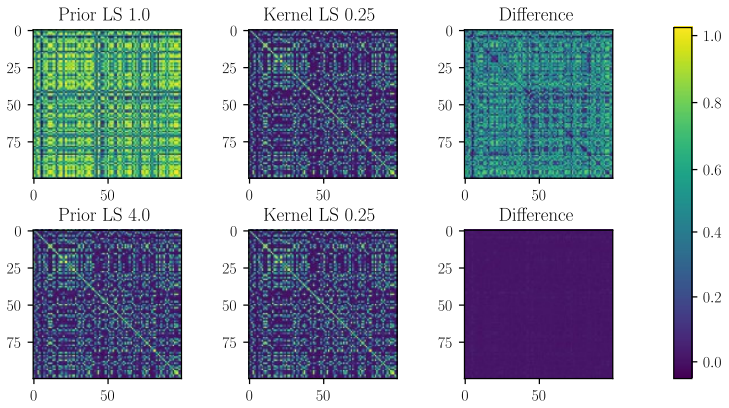

Figure 5: Validation of the effectiveness of the weight-space sampling method on hyperparameter adjusting. The method is evaluated by comparing the difference between the sample covariance matrix obtained using basis functions at different lengthscale settings, 1.0 and 4.0, and the target covariance is computed directly from the standard RBF kernel with lengthscale set to 1/4. The number of samples is 20000, and the number of basis functions is 2048.

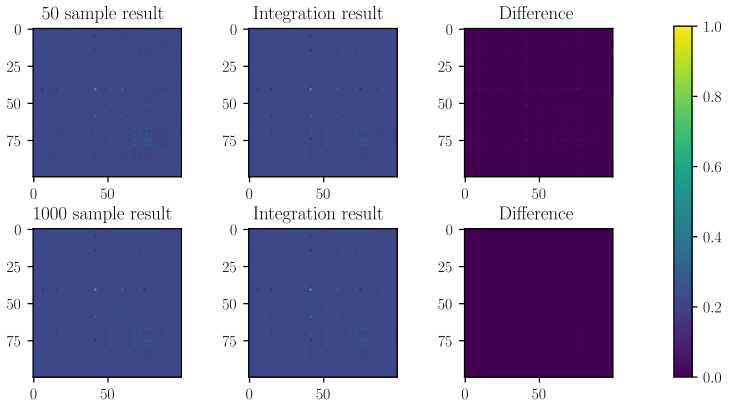

Figure 6: Validation of the effectiveness of the weight-space sampling method on different sample sizes. The method is evaluated by comparing the difference between the sample covariance matrix obtained using Proposition 1 at sample size of 50 and 1000, and the target covariance is computed through the integration $\int p(f|u)q(u)\mathrm{d}u$. The number of basis functions is 2048.

$$
\begin{aligned}
&\int\int_0^{2\pi} \frac{1}{2\pi} p(\theta)\sqrt{2}\cos\left(\theta x + b\right)\sqrt{2}\cos\left(\theta y + b\right)\mathrm{d}\theta\mathrm{d}b \\
&= \int\int_0^{2\pi} \frac{1}{2\pi} p(\theta)\left[\cos\left(\theta(x-y)\right) + \cos\left(\theta(x+y)+2b\right)\right]\mathrm{d}\theta\mathrm{d}b \\
&= k(x,y) + \int\int_0^{2\pi} \frac{1}{2\pi} p(\theta)\cos(\theta(x+y)+2b)\mathrm{d}\theta\mathrm{d}b \\
&= k(x,y).
\end{aligned} \tag{26}
$$

This basis function transformation is known as RFF. The core idea is to replace direct sampling from the covariance with sampling via basis functions, where the choice of distribution for θ depends on the Fourier transform of the kernel. For the RBF kernel, θ follows a standard Gaussian distribution, which leads to efficient computation.

C Feasibility Validation of the Sampling Technique

Although Propositions 1 and 2 provide rigorous mathematical foundations for EDGP, visualizations can further enhance the model's confidence. In this section, we present a set of validation experiments to support the proposed method's feasibility and analyze the impact of its hyperparameters.

We first focus on validating Proposition 2, which concerns whether sampling based on its formulation can successfully restore the covariance structure of the kernel. A related and equally important issue is how to optimize kernel hyperparameters, since Proposition 2 only analyzes the standard RBF kernel without addressing how the associated basis functions adapt when parameters like the lengthscale (LS) change.

Figure 5 addresses this concern. The first row shows a large error, indicating that changes in LS indeed affect the precision of covariance restoration. For example, if the kernel's LS is updated from 1.0 to 0.25 while the LS of the prior $p(\theta)$ remains fixed at 1.0, the sampling method breaks down. This is because the LS of the kernel and that of the prior are reciprocal, as supported by the scaling property of Fourier transform $f(at) \overset{\mathcal{F}}{\rightarrow} \frac{1}{|a|}F(\frac{\omega}{a})$.

Maintaining this reciprocal relationship during training ensures that the sampling remains valid at all times, as demonstrated in the second row of Figure 5.

Next, we verify Proposition 1, which states that this sampling approach should also recover the posterior distribution's covariance. We are particularly interested in how the sampling accuracy depends on the number of samples, since this directly affects computational cost. The goal is to achieve high accuracy with as few samples as possible.

Figure 6 illustrates this relationship. While the restoration accuracy is already quite good with 50 samples, increasing the number to 1000 further reduces the error between the restored covariance and the integrated (ground truth) covariance. Nonetheless, using a smaller number of samples remains a practical and effective choice.

D Derivation of the ELBO

In this section, we derive the ELBO (Equation 15) and show that EDGP, like DSDGP, achieves scalability through data sub-sampling, making it suitable for extremely large datasets. The derivation can begin by minimizing the KL divergence and showing that the sum of the ELBO and the KL divergence equals the marginal log-likelihood. This implies that maximizing the ELBO is equivalent to minimizing the KL divergence. However, in this paper we follow the VFE tradition that directly applies Jensen's inequality to lower-bound the marginal log-likelihood, yielding the ELBO as:

$$\begin{aligned}
&\log p(\mathbf{y}) \\
&= \log \left\{ \int \left[q(\{\mathbf{f}^l, \mathbf{u}^l\}_{l=1}^L) \frac{p(\mathbf{y}|\mathbf{f}^L) \prod_{l=1}^L p(\mathbf{f}^l|\mathbf{u}^l; \mathbf{Z}^l, \mathbf{f}^{l-1}) p(\mathbf{u}^l)}{q(\{\mathbf{f}^l, \mathbf{u}^l\}_{l=1}^L)} \right] \mathrm{d}\mathbf{f}^L \mathrm{d}\mathbf{u}^L \dots \right\} \\
&\geq \int q(\{\mathbf{f}^l, \mathbf{u}^l\}_{l=1}^L) \log \frac{p(\mathbf{y}|\mathbf{f}^L) \prod_{l=1}^L p(\mathbf{f}^l|\mathbf{u}^l; \mathbf{Z}^l, \mathbf{f}^{l-1}) p(\mathbf{u}^l)}{q(\{\mathbf{f}^l, \mathbf{u}^l\}_{l=1}^L)} \mathrm{d}\mathbf{f}^L \mathrm{d}\mathbf{u}^L \dots,
\end{aligned} \tag{27}$$

where in VFE, the first term $\int \left\{ q(\{\mathbf{f}^l, \mathbf{u}^l\}_{l=1}^L) \log p(\mathbf{y}|\mathbf{f}^L) \mathrm{d}\mathbf{f}^L \right\}$ is analyzed to obtain a closed-form solution, and the optimal variational distribution is derived via functional optimization. This has the advantage of introducing a diagonal regularization term into the objective, which helps prevent overfitting.

EDGP, in contrast, does not yield a closed-form solution and instead relies on sampling to compute an unbiased estimator of the objective. Equation 27 can thus be rewritten as:

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{f}^L; \mathbf{Z}^L, \tilde{\mathbf{f}}^{L-1})}[\log p(\mathbf{y}|\mathbf{f}^L)] - \sum_{l=1}^{L} \mathrm{KL}[q(\mathbf{u}^l)||p(\mathbf{u}^l; \mathbf{Z}^l)]. \tag{28}$$

Since the likelihood term $\log p(\mathbf{y}|\mathbf{f}^L)$ factorizes over the data, the estimator can be expressed as:

$$\mathcal{L} = \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{f}_i^L; \mathbf{Z}^L, \tilde{\mathbf{f}}^{L-1})}[\log p(\mathbf{y}_i|\mathbf{f}_i^L)] - \sum_{l=1}^{L} \mathrm{KL}[q(\mathbf{u}^l)||p(\mathbf{u}^l; \mathbf{Z}^l)]. \tag{29}$$

This form allows the model to be trained incrementally via dataset sub-sampling, much like standard neural networks, significantly expanding the range of scenarios where EDGP can be applied. As shown in the experimental results in Section 4.2, EDGP achieves training efficiency nearly on par with neural baselines like FCN and LSTM.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main contribution of this paper is to improve the existing doubly stochastic deep gaussian processes methods. The EDGP proposed in this paper is quicker, and more rigorous and accurate.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have provided such information in the Discussion and Limitations section 5. The main limiration of EDGP is its flexibility in choosing kernel.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The proof is provided in appendix A and B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All experiment details have been disclosed in section 4

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The implementation code of EDGP and DSDGP is in the supplementary material as a zip file. After review, we will provide the GitHub URL.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All experiment details have been disclosed in section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All experimental results were obtained after 20 repetitions. The standard deviation is also presented in section 4 to ensure that the results are significant.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Computer information are disclosed in section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper conforms the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: This paper have used the existing code repository: DSDGP; and dataset: ETTh, Exchange, Debutanizer, SRU, and we have cited the original paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.