Article

Conditional Deep Gaussian Processes: Empirical Bayes Hyperdata Learning

Chi-Ken Lu [1,*] and Patrick Shafto [1,2]

1 Mathematics and Computer Science, Rutgers University, Newark, NJ 07102, USA; patrick.shafto@rutgers.edu
2 School of Mathematics, Institute for Advanced Studies, Princeton, NJ 08540, USA
* Correspondence: cl1178@rutgers.edu

Abstract: It is desirable to combine the expressive power of deep learning with Gaussian Process (GP) in one expressive Bayesian learning model. Deep kernel learning showed success as a deep network used for feature extraction. Then, a GP was used as the function model. Recently, it was suggested that, albeit training with marginal likelihood, the deterministic nature of a feature extractor might lead to overfitting, and replacement with a Bayesian network seemed to cure it. Here, we propose the conditional deep Gaussian process (DGP) in which the intermediate GPs in hierarchical composition are supported by the hyperdata and the exposed GP remains zero mean. Motivated by the inducing points in sparse GP, the hyperdata also play the role of function supports, but are hyperparameters rather than random variables. It follows our previous moment matching approach to approximate the marginal prior for conditional DGP with a GP carrying an effective kernel. Thus, as in empirical Bayes, the hyperdata are learned by optimizing the approximate marginal likelihood which implicitly depends on the hyperdata via the kernel. We show the equivalence with the deep kernel learning in the limit of dense hyperdata in latent space. However, the conditional DGP and the corresponding approximate inference enjoy the benefit of being more Bayesian than deep kernel learning. Preliminary extrapolation results demonstrate expressive power from the depth of hierarchy by exploiting the exact covariance and hyperdata learning, in comparison with GP kernel composition, DGP variational inference and deep kernel learning. We also address the non-Gaussian aspect of our model as well as way of upgrading to a full Bayes inference.

Keywords: deep Gaussian process; approximate inference; deep kernel learning; Bayesian learning; moment matching; inducing points; neural network

1. Introduction

The deep Gaussian process [1] is a Bayesian learning model which combines both the expressive power of deep neural networks [2] and calibrated uncertainty estimation. The hierarchical composition of Gaussian Processes (GPs) [3] is the origin of expressiveness, but also renders inference intractable, as the marginalization of GPs in the stage of computing evidence is not analytically possible. Expectation propagation [4,5] and variational inference [6–9] are approximate inference schemes for DGP. The latter has issues of posterior collapse, which turns DGP into a GP with transformed input. References [8,9] address this issue and compositional freedom [10] in such hierarchical learning. Nevertheless, inferential challenges continue to slow the adoption of DGP.

Despite challenges, there has been progresses in understanding this seemingly simple yet profound model. In the case where the GPs in the hierarchy are zero-mean, DGP exhibits pathology, becoming a constant function as the depth increases [11]. Using the fact that the exponential covariance function is strictly convex, references [12,13] studied the conditional statistics for squared distance in function space, suggesting region in hyperparameter space to avoid the pathology. Recently, reference [14] showed the connection between DGP and a deep neural network with bottlenecked layers, and reference [15] suggested that a DGP with a large width may collapse back to a GP.

Others have found ways to work around the challenges of DGPs. The deep kernel learning proposed in [16] gained the Bayesian character of GP and the expressive power of a deep neural network without encountering intractability, as the learning of weight parameters, treated as kernel hyperparameters, is an empirical Bayes. Similar ideas also appeared in [17,18]. Hyperparameter learning in [16] was performed through marginal likelihood, which can in principle prevent overfitting due to the built-in competition between data fitting and model complexity [3]. However, [19] suggested that the lack of Bayesian character in the deep feature extracting net might still result in overfitting if the network has too many parameters.

Here, we propose a conditional DGP model in which the intermediate GPs (all but the exposed GP) in the hierarchical composition are conditioned on a set of hyperdata. These hyperdata are inspired by the inducing points in sparse GP [20–22], but *they are hyperparameters, not random variables*. The conditional DGP is motivated by the expressive power and Bayesian character of DGP [1], and the deep kernel learning with an objective in marginal likelihood [16]. Due to the conditioning on the hyperdata, the intermediate GPs can be viewed as collections of random feature functions centered around the deterministic conditional mean. Thus, the intermediate GPs become approximately deterministic functions when the hyperdata are sufficiently dense. Besides, lifting the intermediate GPs from being zero mean might help avoid pathology too. Mathematically, we defined a marginal prior for the conditional DGP; i.e., all intermediate GPs are marginalized, which assures the Bayesian character when dealing with the feature functions. We then use the moment matching method to approximate the non-Gaussian marginal prior as a GP [23], which connects with observed data and allows the marginal likelihood objective. It should be stressed that the effective kernel depends on the conditional mean and conditional covariance in feature function via the hyperdata, which are optimized in the spirit of empirical Bayes [24]. In the implementation, the hyperdata supporting each intermediate GP are represented as a neural network function, $u = \mathrm{nn_w}(z)$ with u and z being the output and input of hyperdata, similarly to the trick used in modeling the mean and variance for data in the variational autoencoder [25].

The paper is organized as follows. Section 2 gives a short survey of the current literature on deep probabilistic models; the usage of moment matching in approximate inference; and the inducing points in GP and DGP. Background on mathematical models of GP and DGP, the marginal prior for DGP and the moment matching method are introduced in Section 3. The conditional DGP with SE kernel in the exposed layer, its mathematical connection with deep kernel learning, the parameter learning and the non-Gaussian aspect, are described in Section 4. A preliminary demonstration on extrapolating two time-series data is in Section 5, followed by a discussion in Section 6.

2. Related Work

In the literature on deep probabilistic models, [26] proposed the conditional neural process in which the mean and variance functions are learned from the encoded representation of context data in a regression setup for target data. Deep Gaussian processes (DGPs) constitute one family of models for composition functions by conditioning input to a GP on the output of another GP [1]. A similar idea appeared in the works of warped GP [27,28]. The implicit process in [29] is a stochastic process embedding the Gaussian distribution into a neural network. Solutions of stochastic differential equation driven by GP are also examples of composite processes [30]. Variational DGP casts the inference problem in terms of optimizing ELBO [6] or EP [5]. However, the multi-modalness of DGP posterior [10,23] may arise from the fact that the hidden mappings in intermediate layers are dependent [9]. Inference schemes capable of capturing the multi-modal nature of DGP posterior were recently proposed by [8,9]. Depth of neural network models and the function expressivity were studied in [31,32], and uncertainty estimates were investigated in [33]. DGP in weight space representation and its variational Bayesian approach to DGP inference were introduced in [34], which were based on the notion of random feature

expansion of Gaussian [35] and arcsine [36] kernels. Deep hierarchical SVMs and PCAs were introduced in [37].

Moment matching is a way to approximate a complex distribution with, for instance, a Gaussian by capturing the mean and the second moment. Reference [38] considered a GP regression with uncertain input, and replaced the non-Gaussian predictive distribution with a Gaussian carrying the matched mean and variance. Expectation propagation, in [4], computed the vector of mean and variance parameters of non-Gaussian posterior distributions. Reference [21] approximated the distribution over unseen pixels as a multivariate Gaussian with matched mean and covariance. Moment matching is also extensively applicable to comparing two distributions [39] where the embedded means in RKHS are computed. In generative models, the model parameters are learned from comparing the model and data distributions [40].

Inducing points are an important technique in sparse GP [20,22,41,42] and DGP. In addition to being locally defined as a function's input and output, [43] introduced a transformation to form a global set of inducing *features*. One popular transformation uses the basis of Gaussian so that one can recover the local inducing points easily [43]. Transformation using the basis of spherical harmonic functions in [44] allows orthogonal inducing features and connects with the arcsine kernels of Bayesian deep neural network [45]. Reference [46] employed the inter-domain features in DGP inference. Recently, [47] proposed a method to express the local inducing points in the weight space representation. All the methods cited here treated the inducing points or features in a full Bayes approach, as they are random variables associated with an approximate distribution [24].

3. Background

Here, we briefly introduce the notions of the Gaussian process as a model for random continuous function $f(\mathbf{x}) : \mathbb{R}^d \mapsto \mathbb{R}$. A deep Gaussian process [1] is a hierarchical composition of Gaussian processes for modeling general composite function $\mathbf{f}_L \circ \mathbf{f}_{L-1} \circ \cdots \mathbf{f}_2 \circ \mathbf{f}_1(\mathbf{x})$ where the bold faced function $\mathbf{f}_1 : \mathbb{R}^d \mapsto \mathbb{R}^{H_1}$ has an output consisting of H_1 independent GPs, and similarly for $\mathbf{f}_2 : \mathbb{R}^{H_1} \mapsto \mathbb{R}^{H_2}$ and so on. The depth and width of DGP are thus denoted by L and $H_{1:L}$, respectively.

3.1. Gaussian Process

In machine learning, the attention is often restricted to the finite set of correlated random variables $\mathbf{f} := \{f(\mathbf{x}_1), \cdots, f(\mathbf{x}_N)\}$ corresponding to the design location $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_N)^T$. Denoting $f_i := f(\mathbf{x}_i)$, the above set of random variables is a GP if and only if the following relations,

$$\mathbb{E}[f_i] = \mu(\mathbf{x}_i), \ \mathbb{E}[(f_i - \mu_i)(f_j - \mu_j)] = k(\mathbf{x}_i, \mathbf{x}_j), \tag{1}$$

are satisfied for all indices i, j. For convenience, we can use $f \sim \mathcal{GP}(\mu, k)$ to denote the above. The mean function $\mu(\cdot) : \mathbb{R}^d \mapsto \mathbb{R}$ and the covariance function $k(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ then fully specify the GP. One can proceed to write down the multivariate normal distribution as the pdf

$$p(\mathbf{f}) = \frac{1}{\sqrt{(2\pi)^N |K|}} \exp[-\frac{1}{2}(\mathbf{f} - \mathbf{m})^t K^{-1}(\mathbf{f} - \mathbf{m})]. \tag{2}$$

The covariance matrix K has matrix element $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, characterizing the correlation between the function values. The covariance function k encodes function properties such as smoothness. The vector $\mathbf{m} := \mu(\mathbf{X})$ represents the mean values at corresponding inputs. Popular covariance functions include the squared exponential (SE) $k(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \exp[-||\mathbf{x}_i - \mathbf{x}_j||^2/(2\ell^2)]$ and the family of Matern functions. The signal magnitude σ and length scale ℓ are hyper-parameters.

The conditional property of Gaussians allows one to place constraint on the model $p(\mathbf{f})$. Given a set of function values $\mathbf{u} = f(\mathbf{Z})$, the space of random function f now only

includes those passing through these fixed points. Then the conditional pdf $p(\mathbf{f}|\mathbf{u})$ has the conditional mean and covariance:

$$\mathbf{m} \rightarrow \mathbf{m} + K_{\mathbf{xz}}K_{\mathbf{zz}}^{-1}[\mathbf{u} - \mathbf{m}] \tag{3}$$

$$\mathbf{K_X} \rightarrow \mathbf{K_X} - \mathbf{K_{XZ}}\mathbf{K_Z}^{-1}\mathbf{K_{ZX}} \tag{4}$$

where the matrix $\mathbf{K_{XZ}}$ represents the covariance matrix evaluated at $\mathbf{X}$ against $\mathbf{Z}$.

3.2. Deep Gaussian Process

We follow the seminal work in [1] to generalize the notion of GP to the composite functions $\mathbf{f}_L \circ \mathbf{f}_{L-1} \circ \cdots \mathbf{f}_2 \circ \mathbf{f}_1(\mathbf{x})$. In most literature, DGP is defined from a generative point of view. Namely, the joint distribution for the simplest zero-mean DGP with $L = 2$ and $H_2 = H_1 = 1$ can be expressed as

$$p(f_2, f_1|\mathbf{X}) = p(f_2|f_1)p(f_1|\mathbf{X}), \tag{5}$$

with the conditional defined as $f_2|f_1 \sim \mathcal{GP}(0, k(f_1, f_1))$ and $f_1 \sim \mathcal{GP}(0, k(\mathbf{X}, \mathbf{X}))$.

3.3. Marginal Prior, Covariance and Marginal Likelihood

In the above DGP model, the *exposed* GP for f_2 is connected with the data output $\mathbf{y}$, and the *intermediate* GP for f_1 with the data input $\mathbf{X}$. In Bayesian learning, both fs shall be marginalized when computing the evidence. Now we define the marginal prior as

$$p(\mathbf{f}) = \int d\mathbf{f}_1 \, p(\mathbf{f}_2|\mathbf{f}_1)p(\mathbf{f}_1|\mathbf{X}). \tag{6}$$

in which the bold faced $\mathbf{f}_1$ representing the set of intermediate function values are marginalized, but the exposed $\mathbf{f}_2$ is not. Note that the notation $f(\mathbf{x}) = f_2(f_1(\mathbf{x}))$ is not ambiguous in a generative view, but may cause some confusion in the marginal view as the label f_1 has been integrated out. To avoid confusing with the exposed function $f_2(\cdot)$, we still use $f(\cdot)$ to denote the marginalized composite function unless otherwise stated.

Motivated to write down an objective in terms of marginal likelihood, the moment matching method in [23] was proposed, so one can approximate Equation (6) with a multivariate Gaussian $q(\mathbf{f}|\mathbf{X})$ such that the mean and the covariance are matched. In the zero-mean DGP considered in [23], the covariance matching refers to

$$\mathbb{E}_{\mathbf{f}\sim q}[f_i f_j] = \mathbb{E}_{\mathbf{f}_1}[\mathbb{E}_{\mathbf{f}_2|\mathbf{f}_1}[f_i f_j]] = \int d\mathbf{f}_2 d\mathbf{f}_1 f_2(f_1(\mathbf{x}_i))f_2(f_1(\mathbf{x}_j))p(\mathbf{f}_2|\mathbf{f}_1)p(\mathbf{f}_1|\mathbf{X}). \tag{7}$$

In the case where the squared exponential kernel is used in both GPs, the approximate marginal prior $q(\mathbf{f}|\mathbf{X}) = \mathcal{N}(0, K_{\text{eff}})$, with the effective kernel being $k_{\text{eff}} = \sigma_2^2[1 + 2\frac{\sigma_1^2}{\ell_2^2}(1 - \exp(-|\mathbf{x}_i - \mathbf{x}_j|^2/2\ell_1^2))]^{-\frac{1}{2}}$ [23]. The hyperparameters include the length scale ℓ and signal magnitude σ with layer indexed at the subscript.

Consequently, the evidence of the data $\mathbf{X}, \mathbf{y}$ associated with the 2-layer DGP can be approximately expressed as

$$p(\mathbf{y}|\mathbf{X}) \approx \int d\mathbf{f} p(\mathbf{y}|\mathbf{f})q(\mathbf{f}|\mathbf{X}). \tag{8}$$

Thus, the learning of hyperparameters σs and ℓs in the zero-mean DGP model is through the gradient descent on $\log p(\mathbf{y}|\mathbf{X})$, and the gradient components $\frac{\partial K}{\partial \ell_{1,2}}$ and $\frac{\partial K}{\partial \sigma_{1,2}}$ are needed in the framework of GPy [48].

4. Model

Following the previous discussion, we shall introduce the model of conditional DGP along with the covariance and marginal prior. The mathematical connection with deep

kernel learning and the non-Gaussian aspect of marginal prior will be discussed. The difference between the original DGP and the conditional DGP is that the intermediate GPs in the latter are conditioned on the hyperdata. Learning the hyperdata via the approximate marginal likelihood is, loosely speaking, an empirical Bayesian learning of the feature function in the setting of deep kernel learning.

4.1. Conditional Deep Gaussian Process

In the simple two-layer hierarchy with width $H_1 = H_2 = 1$, the hyperdata $\{\mathbf{Z}, \mathbf{u}\} = \{\mathbf{z}_{1:M} \in \mathbb{R}^d, u_{1:M} \in \mathbb{R}\}$ are introduced as support for the intermediate GP for f_1, and the exposed GP for f_2 remains zero-mean and does not condition on any point. Thus, f_1 can be viewed as a space of random functions constrained with $f_1(\mathbf{z}_{1:M}) = u_{1:M}$, and the Gaussian distribution $p(f_1(\mathbf{x}_{1:N})|\mathbf{Z}, \mathbf{u})$ has its conditional mean and covariance in Equation (3) (with **m** on RHS set to zero) and (4), respectively. Following Equation (6), the marginal prior for this conditional DGP can be similarly expressed as

$$p(\mathbf{f}) = \int d\mathbf{f}_1 p(\mathbf{f}_2|\mathbf{f}_1) p(\mathbf{f}_1|\mathbf{X}, \mathbf{Z}, \mathbf{u}) \,. \tag{9}$$

With f_1 being conditioned on the hyperdata $\{\mathbf{Z}, \mathbf{u}\}$, one can see that the multivariate Gaussian $p(f_1(\mathbf{x}_{1:N})|\mathbf{Z}, \mathbf{u})$ emits samples in the space of random functions passing through the fixed hyperdata so that Equation (9) is a sum of an infinite number of GPs. Namely,

$$f \sim \sum_{f_1} \mathcal{GP}(0, k_2(f_1(\mathbf{X}), f_1(\mathbf{X}))) \,,$$

with f_1 under the constraints due to the hyperdata and the smoothness implied in kernel k_1. Therefore, f are represented by an ensemble of GPs with same kernel but different *feature* functions. We shall come back to this point more rigorously in Section 4.2.

Now we shall approximate the intractable distribution in Equation (9) with a multivariate Gaussian $q(\mathbf{f}|\mathbf{X}, \mathbf{Z}, \mathbf{u})$ carrying the matched covariance. The following lemma is useful for the case where the exposed GP for $f_2|f_1$ uses the squared exponential (SE) kernel.

Lemma 1. *(Lemma 3 in [49]) The covariance in $p(\mathbf{f})$ (Equation (9)) with the SE kernel $k_2(x, y) = \sigma_2^2 \exp[-(x-y)^2/2\ell_2^2]$ in the exposed GP for $f_2|f_1$ can be calculated analytically. With the Gaussian conditional distribution, $p(\mathbf{f}_1|\mathbf{X}, \mathbf{Z}, \mathbf{u})$, supported by the hyperdata, the effective kernel reads*

$$k_{\text{eff}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sigma_2^2}{\sqrt{1 + \delta_{ij}^2/\ell_2^2}} \exp\left[-\frac{(m_i - m_j)^2}{2(\ell_2^2 + \delta_{ij}^2)} \right], \tag{10}$$

where $m_{i,j} := m(\mathbf{x}_{i,j})$ and $c_{ij} := \text{cov}(f_1(\mathbf{x}_i), f_1(\mathbf{x}_j))$ are the conditional mean and covariance, respectively, at the inputs $\mathbf{x}_{i,j}$. The positive parameter $\delta_{ij}^2 := c_{ii} + c_{jj} - 2c_{ij}$ and the the length scale ℓ_2 dictates how the uncertainty about f_1 affects the function composition.

Next, in addition to the hyperparameters such as σs and ℓs, the function values $u_{1:M}$ are hyperdata that shall be learned from the objective. With approximating the non-Gaussian marginal prior $p(\mathbf{f}|\mathbf{X}, \mathbf{Z}, \mathbf{u})$ with $q(\mathbf{f}|\mathbf{X}, \mathbf{Z}, \mathbf{u})$, we are able to compute the approximate marginal likelihood as the objective

$$\mathcal{L} = -\log \int d\mathbf{f} p(\mathbf{y}|\mathbf{f}) q(\mathbf{f}|\mathbf{X}, \mathbf{Z}, \mathbf{u}) \,. \tag{11}$$

The learning of all hyperparameter data follows the standard gradient descent used in GPy [48], and the gradient components include the usual ones, such as $\partial K_{\text{eff}}/\partial \ell_2$ in exposed layer and those related to the intermediate layer $\partial K_{\text{eff}}/\partial \ell_1$ and the hyperdata $\partial K_{\text{eff}}/\partial u_{1:M}$ through chaining with $\partial K_{\text{eff}}/\partial(m_i - m_j)$ and $\partial K_{\text{eff}}/\partial \delta_{ij}^2$ via Equations (3) and (4). To exploit

the expressive power of neural network during optimization, the hyperdata can be further modeled by a neural network; i.e.,

$$u_{1:M} = \text{nn}_{\mathbf{w}}(\mathbf{z}_{1:M}),\tag{12}$$

with $\mathbf{w}$ denoting the weight parameters. In such case, the weights $\mathbf{w}$ are learned instead of the hyperdata $u_{1:M}$.

4.2. When Conditional DGP Is Almost a GP

In the limiting case where the probabilistic nature of f_1 is negligible, then the conditional DGP becomes a GP with the transformed input; i.e., the distribution $p(\mathbf{f}_1|\mathbf{X}, \mathbf{Z}, \mathbf{u})$ becomes highly concentrated around a certain conditional mean $\bar{f}_1(\mathbf{x})$. To get insight, we reexamine the covariance in the setting where f_1 is almost deterministic. We can reparameterize the random function f_1 at two distinct inputs $\mathbf{x}_{1,2}$ for the purpose of computing covariance:

$$f_1(\mathbf{x}_{i,j}) = m(\mathbf{x}_{i,j}) + \epsilon_{i,j},\tag{13}$$

where $m(\mathbf{x})$ is the conditional mean given the fixed $\mathbf{Z}$ and $\mathbf{u}$. The random character lies in the two correlated random variables, $(\epsilon_i, \epsilon_j)^T \sim \mathcal{N}(0, C)$, corresponding to the weak but correlated signal around zero. Under that assumption, we follow the analysis in [9,38] and prove the following lemma.

Lemma 2. *Consider $p(f)$ defined in Equation (9), with $f_2|f_1$ being a more general $\mathcal{GP}(\mu_2, k_2)$ and f_1 reparametrized as in Equation (13). The covariance, $\text{cov}(f_2(f_1(\mathbf{x}_i)), f_2(f_1(\mathbf{x}_j)))$, has the following form:*

$$[1 + \frac{c_{ii}}{2}\partial^2_{m_j} + \frac{c_{jj}}{2}\partial^2_{m_i} + c_{ij}\partial^2_{m_i m_j}]k_2(m_i, m_j) + c_{ij}\mu'_2(m_i)\mu'_2(m_j),\tag{14}$$

where the cs are matrix elements of kernel matrix C associated with the weak random variables $\epsilon_{i,j}$ in Equation (13). The notation $m_{i,j} := m(\mathbf{x}_{i,j})$ and prime as derivative is used.

Proof. The assumption is that $f_2|f_1 \sim \mathcal{GP}(\mu_2, k_2)$ and that $f_1(x)$ a weak random function $\epsilon(x)$ overlaying a fixed function $m(x)$. At any two inputs $x_{i,j}$, we expand the target function f to the second order:

$$f(x_{i,j}) = f_2(f_1(x_{i,j})) \approx f_2(m_{i,j}) + \epsilon_{i,j}f'_2(m_{i,j}) + \frac{\epsilon^2_{i,j}}{2}f''_2(m_{i,j}),\tag{15}$$

where the shorthanded notation $m_i := m(x_i)$ and $\epsilon(x_i) := \epsilon_i$ is used. Note that (ϵ_i, ϵ_j) is bivariate Gaussian with zero mean and covariance matrix $\mathbf{C}$. We use the law of total covariance, $\text{cov}[a, b] = \text{cov}(\mathbb{E}[a|d], \mathbb{E}[b|d]) + \mathbb{E}[\text{cov}(a, b|d)]$ with a, b and d being some random variables. To proceed with the first term, we calculate the conditional mean given the ϵs:

$$\mathbb{E}[f(x_{i,j})|\epsilon_i, \epsilon_j] = \mu_2(m_{i,j}) + \epsilon_{i,j}\mu'(m_{i,j}).\tag{16}$$

Then one uses the fact that $f_2|f_1$, $f'_2|f_1$ and $f''_2|f_1$ are jointly Gaussian to compute the conditional covariance, which can be expressed in a compact form:

$$\text{cov}[f(x_i)f(x_j)|\epsilon_i, \epsilon_j] = \hat{O}(\epsilon_i, \epsilon_j)k(m_i, m_j).\tag{17}$$

The operator $\hat{O}$ accounts for the fact that $\text{cov}[f'_2(m_i), f'_2(m_j)] = \partial^2_{m_i m_j}k_2(m_i, m_j)$ and $\text{cov}[f_2(m_i), f''_2(m_j)] = \partial^2_{m_j}k_2(m_i, m_j)$. Thus, the operator reads

$$\hat{O} = 1 + \epsilon_i\partial_{m_i} + \epsilon_j\partial_{m_j} + \frac{\epsilon^2_i}{2}\partial^2_{m_i} + \frac{\epsilon^2_j}{2}\partial^2_{m_j} + \epsilon_i\epsilon_j\partial^2_{m_i m_j}.\tag{18}$$

Now we are ready to deal with the outer expectation with respect to the ϵs. Note that the covariance $c_{ij} := \mathbb{E}[\epsilon_i \epsilon_j] = c(x_i, x_j)$ and variance $c_{ii} := \mathbb{E}[\epsilon_i^2] = c(x_i, x_i)$ are matrix elements of $\mathbf{C}$. Consequently, we prove the total covariance in Equation (14). $\square$

Remark 1. *Since the second derivatives* $\partial_{m_i}^2 k_2(m_i, m_j) = \partial_{m_j}^2 k_2(m_i, m_j) = -\partial_{m_i m_j}^2 k_2(m_i, m_j)$ *hold for the stationary* k_2*, the above covariance (Equation (14)) with* $\mu_2 = 0$ *is identical to the effective kernel in Equation (10) in the limit* $\ell_2^2 \gg \delta_{ij}^2$*, which reads*

$$\text{cov}(f(x_i)f(x_j)) \propto [1 + \frac{(m_i - m_j)^2 - \ell_2^2}{2\ell_2^4} \delta_{ij}^2] \exp[-\frac{(m_i - m_j)^2}{2\ell_2^2}]. \tag{19}$$

Such a situation occurs when the inputs $\mathbf{Z}$ *in hyperdata are dense enough so that* f_1 *becomes almost deterministic.*

Consequently, in the limit when the conditional covariance in δ^2 is small compared with the length scale ℓ_2^2, Equation (19) indicates that the effective kernel is the SE kernel with a deterministic input $m(\mathbf{x})$, which is equivalent to the deep kernel with SE as the base kernel (see Equation (5) in [16]). On the other hand, when δ^2 and ℓ_2 are comparable, the terms within the first bracket in the RHS of Equation (19) are a non-stationary function which may attribute multiple frequencies in the function f. The deep kernel with the spectral mixture kernel (Equation (6) in [16]) as the base is similar to the effective kernel.

4.3. Non-Gaussian Aspect

The statistics of the non-Gaussian marginal prior $p(\mathbf{f}|\mathbf{X}, \mathbf{Z}, \mathbf{u})$ are not solely determined by the moments up to the second order. The fourth moment can be derived in a similar manner in [23] with the help of the theorem in [50]. Relevant discussion of the heavy-tailed character in Bayesian deep neural network can be found in [51–53]. See Lemma A1 for the details of computing the general fourth moment in the case where SE kernel is used in $f_2|f_1$ in the conditional 2-layer DGP. Here, we briefly discuss the non-Gaussian aspect, focusing on the variance of covariance, i.e., by comparing $\mathbb{E}_p[(f(\mathbf{x}_i)f(\mathbf{x}_j))^2]$ and $\mathbb{E}_q[(f(\mathbf{x}_i)f(\mathbf{x}_j))^2]$, with p being the true distribution (Equation (9)) and q being the approximating Gaussian.

In the SE case, one can verify the difference in the fourth order expectation value:

$$\mathbb{E}_p[(f(\mathbf{x}_i)f(\mathbf{x}_j))^2] - \mathbb{E}_q[(f(\mathbf{x}_i)f(\mathbf{x}_j))^2] = \frac{e^{-\frac{(m_i - m_j)^2}{1 + 2\delta_{ij}^2}}}{[1 + 2\delta_{ij}^2]^{1/2}} - \frac{e^{-\frac{(m_i - m_j)^2}{1 + \delta_{ij}^2}}}{[1 + \delta_{ij}^2]} \geq 0,$$

where we have used the fact that the inequalities $(1 + 2\delta^2)^{-1/2} \geq (1 + \delta^2)^{-1}$ and $\exp[-(1 + 2\delta^2)^{-1}] \geq \exp[-(1 + \delta^2)^{-1}]$ hold. Therefore, the inequality suggests the heavy-tailed statistics of the marginal prior $p(f(\mathbf{x}_i), f(\mathbf{x}_j))$ over any pair of function values.

5. Results

The works in [54,55] demonstrate that GPs can still have superior expressive power and generalization if the kernels are dedicatedly designed. With the belief that deeper models generalize better than the shallower counterparts [56], DGP models are expected to perform better in fitting and generalization than GP models do if the same kernel is used in both. However, such expectation may not be fully realized, as the approximate inference may lose some power in DGP. For instance, diminishing variance in the posterior over the latent function was reported in [9] regarding the variational inference for DGP [6]. Here, with a demonstration of extrapolating real-world time series data with the conditional DGP, we shall show that the depth, along with optimizing the hyperdata, does enhance the expressive power and the generalization due to the multiple length scale and multiple-frequency character of the effective kernel. In addition, the moment matching method as

an approximate inference for conditional DGP does not suffer from the posterior collapse. The simulation codes can be found in the github repository.

5.1. Mauna Loa Data

Figure 1a,b shows fitting and extrapolating the classic carbon dioxide data (yellow marks for training, red for test) with GPs using, respectively, the SE kernel and a mixture of SE, periodic SE and rational quadratic kernels [3].

$$k_{mix}(t, t') = \theta_1^2 \exp[-\frac{(t-t')^2}{\theta_2^2}] + \theta_3^2 \exp[-\frac{\sin^2(t-t')}{\theta_4^2}] + \theta_5^2[1 + \frac{(t-t')^2}{\theta_6^2}]^{-\theta_7} .$$

All the θs are hyperparameters in the mixture kernel. As a result of the multiple time scales appearing in the data, the vanilla GP fails to capture the short time trend, but the GP with mixture of kernels can still present excellent expressivity and generalization. The log marginal likelihoods (logML) were 144 and 459 for the vanilla GP and kernel mixture GP, respectively. The two-layer zero-mean DGP with SE kernel in both layers performed better than the single-layer counterpart. In Figure 1c, the GP with the SE[SE] effective kernel has excellent fitting with the training data but has extrapolated poorly. The good fitting may have resulted from the fact that the SE[SE] kernel does capture the character of multiple length scales in DGP. The logML for the SE[SE] GP is 338.
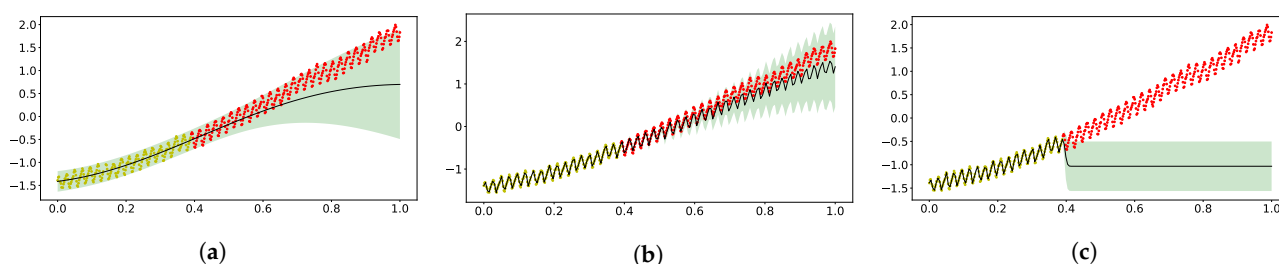

Figure 1. Extrapolation of standardized CO_2 time series data (yellow dots for training and red dots for test) using GP with three kernels. The dark solid line represents the predictive mean, and the shaded area is the the model's confidence. Panel (**a**) displays the result using a single GP with an SE kernel. Panel (**b**) was obtained following the kernel composition in [3]. Panel (**c**) came from using the effective kernel of 2-layer zero-mean DGP with SE used in both layers [23]. (**a**) SE kernel; (**b**) SE+periodic SE+RQ kernel; (**c**) SE[SE] kernel.

Next, we shall see whether improved extrapolation can arise in other deep models or other inference schemes. In Figure 2, the results from DKL and from DGP using the variational inference are shown. Both were implemented in GPFlux [57]. We modified the tutorial code for hybrid GP with three-layer neural network as the code for DKL. The result in Figure 2a does not show good fitting nor good extrapolation, which is to some extent consistent with the simulation of a Bayesian neural network with ReLu activation [32]. As for the DGP using variational inference, the deeper models do not show much improvement compared to the vanilla GP, and the obtained ELBO was 135 for the two-layer DGP (Figure 2b), and it was 127 for three-layer (Figure 2c).
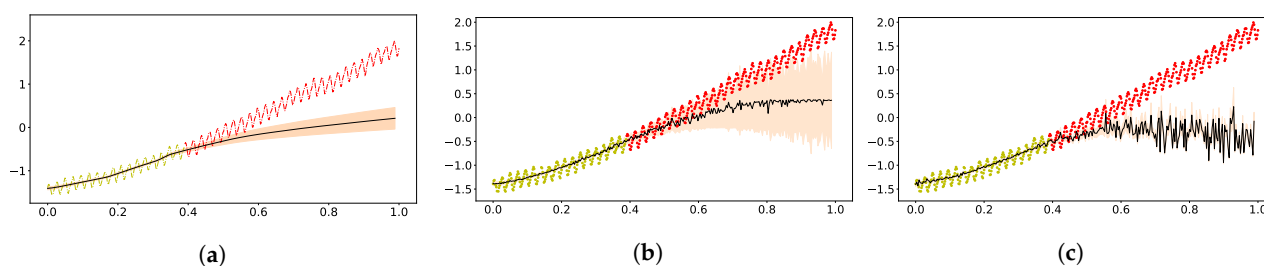
Figure 2. Extrapolation of standardized CO_2 using DKL and variational inference [6] for the DGP implemented in GPFlux [57]. Panel (**a**) was obtained using the DKL with three-layer RELU network. Panel (**b**) shows the results from the two-layer zero-mean DGP model. Panel (**c**) shows the results of the three-layer zero-mean DGP. (**a**) DKL; (**b**) Two-layer DGP; (**c**) Three-layer DGP.

Now we continue to show the performance of our model. In the two-layer model, we have 50 points in hyperdata supporting the intermediate GP. A width-5 tanh neural network is used to represent the hyperdata, i.e., $u_{1:50} = \text{nn}_{\mathbf{w}}(z_{1:50})$. Then, the hyperparameters, including $\sigma_{1,2}$, $\ell_{1,2}$ and weight parameter $\mathbf{w}$, were learned from gradient descent upon the approximate marginal likelihood. The top panel in Figure 3a displays the prediction and confidence from the posterior over f_1, obtained from a GP conditioned on the learned hyperparameters and hyperdata. The logML of the two-layer model was also 338, the same as the SE[SE] GP, and in the bottom panel of Figure 3a one can observe a good fit with the training data. More importantly, the extrapolation shows some high frequency signal in the confidence (shaded region). In comparing ot with Figure 3 of [54], the high-frequency signal only appeared after a periodic kernel was inserted. We attribute the high-frequency signal to the propagation of uncertainty in f_1 (top panel) to the exposed layer (see discussion in Section 4.2).

Lastly, the three-layer model using 37 and 23 hyperdata in the f_1 and f_2 layer, respectively, has its results in Figure 3b. Those hyperdata were parameterized by the same neural network used in the two-layer model. The training had a logML of 253, resulting in a good fit with the data. The extrapolation captured the long term trend in its predictive mean, and the test data were mostly covered in the confidence region. In the latent layers, more expressive patterns overlaying the latent mappings seemed to emerge due to the uncertainty and the depth of the model. The learned $\sigma_{1,2,3} \approx (0.49, 0.86, 2.56)$ and $\ell_{1,2,3} \approx (0.014, 1.2, 0.46)$ show that different layers managed to learn different resolutions.
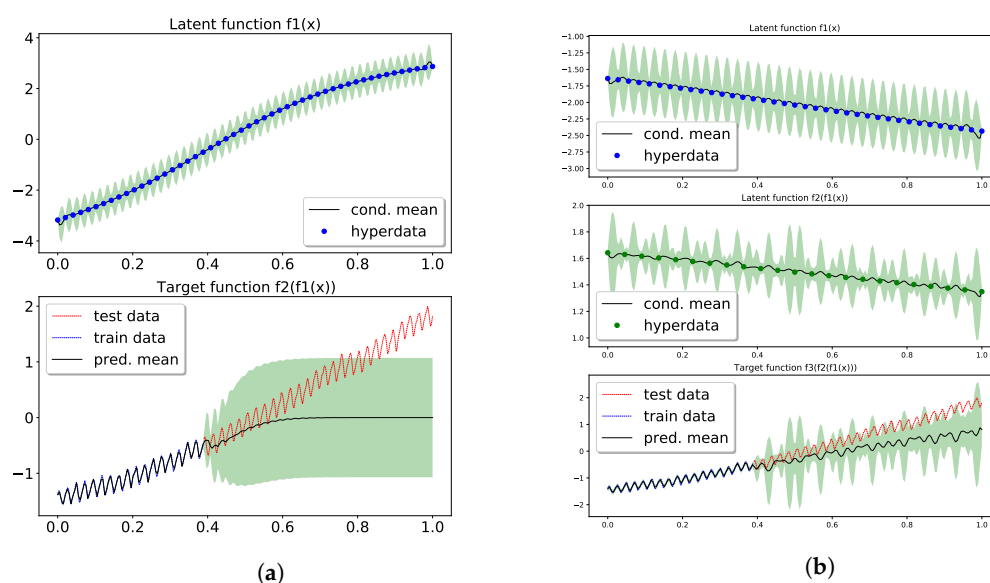

Figure 3. Extrapolation of the standardized CO_2 using conditional DGP. Panel (**a**) is for the two-layer model, and (**b**) for the three-layer model. Top and middle panels shows the mean and confidence in the posterior over the latent functions. See text for details. (**a**) Two-layer conditional DGP; (**b**) Three-layer conditional DGP.

5.2. Airline Data

The models under consideration can be applied to the airline data too. It can be seen in Figure 4 that the vanilla GP was too simple for the complex time-series data, and the GP with the same kernel composition could both fit and extrapolate well. The logML values were -11.7 and 81.9 for the vanilla GP and kernel mixture GP, respectively. Similarly, the SE[SE] kernel captured the multiple length scale character in the data, resulting in a good fit with logML 20.9, but poor extrapolation.
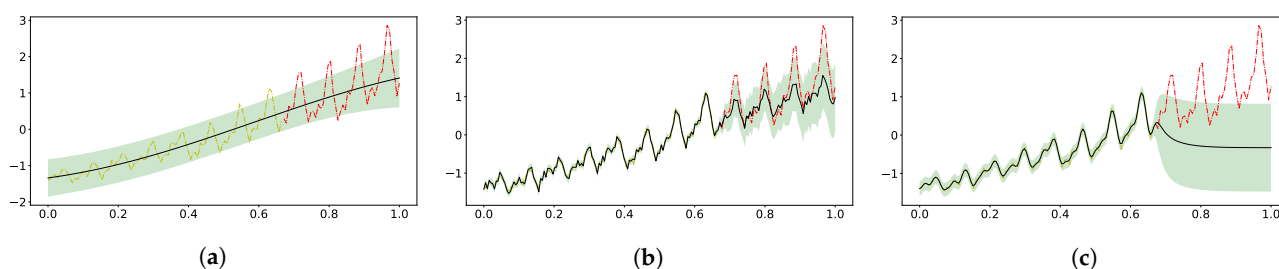

(a)　　　　　　　　　　　　　　(b)　　　　　　　　　　　　　　(c)

Figure 4. Extrapolation of the standardized airline data with three different GPs. (**a**) SE kernel; (**b**) SE+periodic SE+RQ kernel; (**c**) SE[SE] kernel

Here, we display the results using the DKL, variational inference DGP, both two-layer and three-layer, in Figure 5. For the airline data, the DKL with a ReLu neural network as feature extractor panel (Figure 5a) had similar performance to its counterpart on CO_2 data, as did the variational DGP(Figure 5b,c).
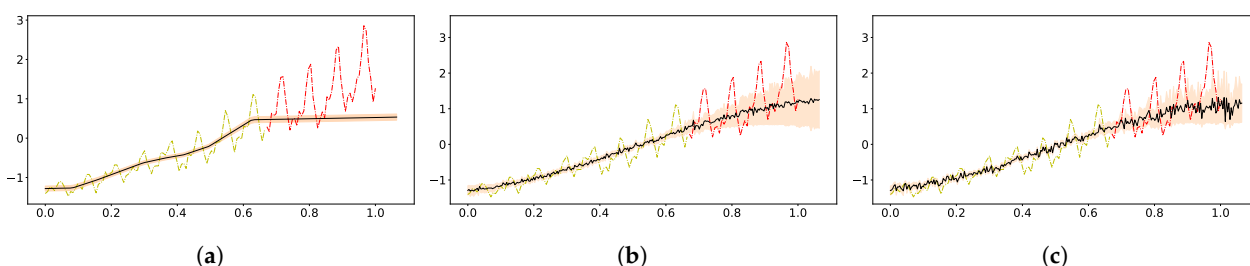

(a)　　　　　　　　　　　　　　(b)　　　　　　　　　　　　　　(c)

Figure 5. Extrapolation of the standardized airline data using DKL (**a**), 2-layer DGP (**b**) and 3-layer DGP (**c**). (**a**) DKL; (**b**) Two-layer DGP; (**c**) Three-layer DGP.

Our two-layer model, aided by the probabilistic latent layer supported by 13 hyperdata, showed improved extrapolation along with the high-frequency signal in prediction and confidence. The optimal logML was 28.5, along with the learned $\sigma_{1,2} = (3.03, 0.73)$, $\ell_{1,2} = (0.026, 2.19)$ and noise level $\sigma_n = 0.004$. As shown in Figure 6b, the latent function supported by learned hyperdata shows an increasing trend on top of an oscillating pattern, which led to the periodic extrapolation in the predictive distribution, albeit only the vanilla kernels were used. It is interesting to compare Figure 6b with Figure 6a, as the latter model had 23 hyperdata supporting the latent function, and the vanishing uncertainty learned in the latent function produced an extrapolation that collapsed to zero. The logML in Figure 6a is 7.25 with learned $\sigma_{1,2} = (2.18, 0.59)$, $\ell_{1,2} = (0.3, 0.07)$, and noise level $\sigma_n = 0.02$.
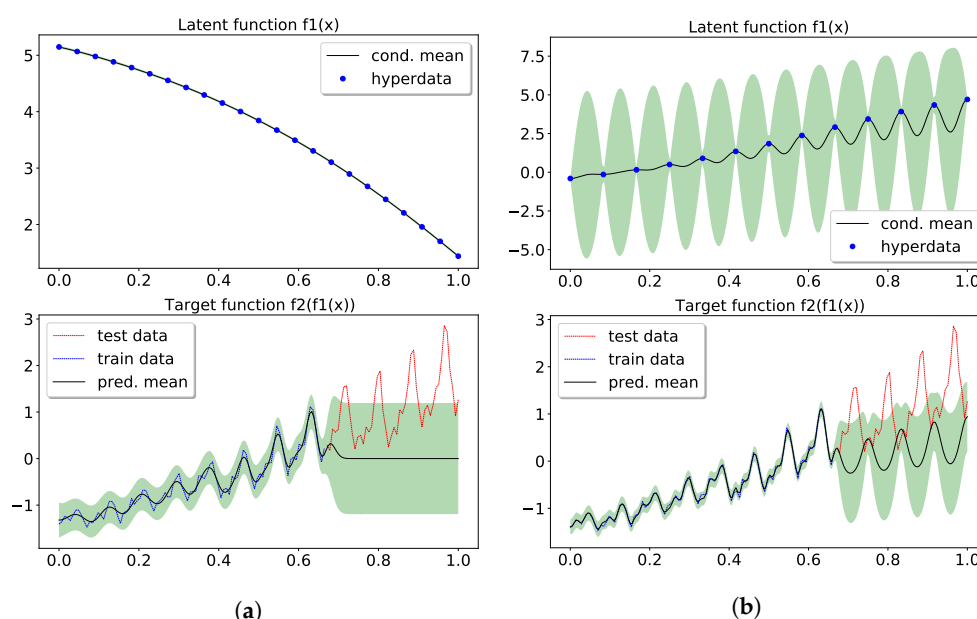
Figure 6. Extrapolation of airline data using conditional DGP. The upper panel shows the learned latent function and uncertainty from hyperdata learning, and the bottom panel shows the extrapolation from the past data. (**a**) The first model had 23 hyperdata supporting the latent GP. (**b**) The other model had 13. (**a**) 2-layer cDGP with 23 hyperdata; (**b**) 2-layer cDGP with 13 hyperdata.

6. Discussion

What did we gain and lose while modifying the original DGP defined in Equation (6) by additionally conditioning the intermediate GPs on the hyperdata? On one hand, when the hyperdata are dense, the conditional DGP is mathematically connected with the deep kernel learning, i.e., a GP with warped input. On the other hand, in the situations when less dense hyperdata are present and the latent GPs are representations of random functions passing through the hyperdata, the conditional DGP can be viewed as an ensemble of deep kernels, and the moment matching method allows us to express it in a closed form. What do we lose in such an approximation? Apparently, the approximate q for the true marginal prior p in Equation (9) cannot account for the heavy-tailed statistics.

In the demonstration, the presence of hyperdata constrains the space of the intermediate functions and moves the mass of the function distribution toward the more probable ones in the process of optimization. Comparing the SE[SE] GP, which represents an approximate version of zero-mean 2-layer DGP, against the conditional DGP model, the constrained space of intermediate functions does not affect the learning significantly, and the generalization is improved. Besides, the uncertainty in the latent layers is not collapsed.

One possible criticism of the present model may result from the empirical Bayes learning of the weight parameters. Although the weight parameters are hyperparameters in both our model and in DKL, it is important to distinguish that the weight parameters in our model parameterize $u_{1:M}$, which supports the intermediate GP, representing an ensemble of latent functions. In DKL, however, the weight parameters fully determine the one latent function, which might lead to overfitting even though marginal likelihood is used as an objective [19]. A possible extension is to consider upgrading the hyperdata to random variables, and the associated mean and variance in $q(u_{1:M})$ can also be modeled as neural network functions of $\mathbf{Z}$. The moment matching can then be applied to approximate the marginal prior $\int d\mathbf{f}_1 d\mathbf{u}\, p(\mathbf{f}_2|\mathbf{f}_1)p(\mathbf{f}_1|\mathbf{X}, \mathbf{Z}, \mathbf{u})q(\mathbf{u})$.

7. Conclusions

Deep Gaussian processes (DGPs), based on nested Gaussian processes (GPs), offer the possibility of expressive inference and calibrated uncertainty, but are limited by intractable marginalization. Approximate inference for DGPs via inducing points and variational infer-

ence allows scalable inference, but incurs costs by limiting expressiveness and causing an inability to propagate uncertainty. We introduce effectively deep kernels with optimizable hyperdata supporting latent GPs via a moment-matching approximation. The approach allows joint optimization of hyperdata and GP parameters via maximization of marginal likelihood. We show that the approach avoids mode collapse, connects DGPs and deep kernel learning, effectively propagating uncertainty. Future directions on conditional DGP include that consideration of randomness in the hyperdata and the corresponding inference.

Author Contributions: Conceptualization, C.-K.L. and P.S.; methodology, C.-K.L.; software, C.-K.L.; validation, C.-K.L. and P.S.; formal analysis, C.-K.L.; investigation, C.-K.L.; resources, C.-K.L.; data curation, C.-K.L.; writing—original draft preparation, C.-K.L.; writing—review and editing, C.-K.L. and P.S.; visualization, C.-K.L.; supervision, P.S.; project administration, C.-K.L.; funding acquisition, P.S. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

GP   Gaussian Process
DGP   Deep Gaussian Process
DKL   Deep Kernel Learning
SE   Squared Exponential

Appendix A

Lemma A1. *Consider the marginal prior for 2-layer conditional DGP [Equation (9)] with $f_2|f_1$ being a GP with SE kernel, and $f_1|\mathbf{Z}, \mathbf{u}$ being another GP with conditional mean μ and conditional covariance k. The general fourth moment is the following sum over distinct doublet decomposition,*

$$\mathbb{E}[f(\mathbf{x}_i)f(\mathbf{x}_j)f(\mathbf{x}_m)f(\mathbf{x}_l)] = \sigma_2^4 \sum \frac{\alpha_{ab,cd}\alpha_{cd,ab}\beta_{ab,cd}}{\sqrt{D_{ab}D_{cd} - V_{ab,cd}^2}}$$

with $V_{ab,cd} = (k_{ad} + k_{bc} - k_{ac} - k_{bd})/\ell_2^2$ and $D_{ab} = 1 + (k_{aa} + k_{bb} - 2k_{ab})/\ell_2^2$. Furthermore, the expressions,

$$\alpha_{ab,cd} = \exp\left[\frac{-(m_a - m_b)^2}{2\ell_2^2(D_{ab} - V_{ab,cd}^2/D_{cd})}\right],$$

and

$$\beta_{ab,cd} = \exp\left[-\frac{(m_a - m_b)(m_c - m_d)V_{ab,cd}}{\ell_2^2(D_{ab}D_{cd} - V_{ab,cd}^2)}\right].$$

Proof. Denoting the function value $h_a := f_1(\mathbf{x}_a)$, we can rewrite the product of the covariance function $k_2(h_a, h_b)k_2(h_c, h_d) = \exp\left(-\frac{[\mathbf{h}]_{ab,cd}^T \mathbb{J}_4 [\mathbf{h}]_{ab,cd}}{2}\right)$ where the row vector $[\mathbf{h}]_{ab,cd}^T = (h_a, h_b, h_c, h_d)$ and the matrix

$$\mathbb{J}_4 = \begin{pmatrix} \mathbb{J}_2 & 0 \\ 0 & \mathbb{J}_2 \end{pmatrix},$$

where $\mathbb{J}_2$ is the 2-by-2 matrix with ones in the diagonal and minus ones in the off-diagonal. The above zeros stand for 2-by-2 zero matrices in the off-diagonal blocks. The procedure of obtaining expectation value with respect to the 4-variable multivariate Gaussian distribution $\mathcal{N}([\mathbf{h}]_{ab,cd}|\mathbb{V}_4, \mathbb{K}_4)$ is similar to the previous one in obtaining the second moment. Namely, applying Lemma 2 in [23],

$$\mathbb{E}[k_2(h_a, h_b)k_2(h_c, h_d)] = \frac{\exp(-\frac{1}{2}\mathbb{V}_4^t \mathbb{A}_4 \mathbb{V}_4)}{\sqrt{I_4 + \mathbb{K}_4 \mathbb{J}_4}},$$

in which the calculation of inverse of 4-by-4 matrix $I_4 + \mathbb{K}_4 \mathbb{J}_4$ and its determinant is quite tedious but tractable. $\square$

References

1. Damianou, A.; Lawrence, N. Deep gaussian processes. In Proceedings of the Artificial Intelligence and Statistics, Scottsdale, AZ, USA, 29 April–1 May 2013; pp. 207–215.
2. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
3. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Process for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006.
4. Minka, T.P. Expectation propagation for approximate Bayesian inference. In Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, Seattle, WA, USA, 2–5 August 2001; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 2001; pp. 362–369.
5. Bui, T.; Hernández-Lobato, D.; Hernandez-Lobato, J.; Li, Y.; Turner, R. Deep Gaussian processes for regression using approximate expectation propagation. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 1472–1481.
6. Salimbeni, H.; Deisenroth, M. Doubly stochastic variational inference for deep gaussian processes. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
7. Salimbeni, H.; Dutordoir, V.; Hensman, J.; Deisenroth, M.P. Deep Gaussian Processes with Importance-Weighted Variational Inference. *arXiv* **2019**, arXiv:1905.05435.
8. Yu, H.; Chen, Y.; Low, B.K.H.; Jaillet, P.; Dai, Z. Implicit Posterior Variational Inference for Deep Gaussian Processes. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 14502–14513.
9. Ustyuzhaninov, I.; Kazlauskaite, I.; Kaiser, M.; Bodin, E.; Campbell, N.; Ek, C.H. Compositional uncertainty in deep Gaussian processes. In Proceedings of the Conference on Uncertainty in Artificial Intelligence, Virtual, 3–6 August 2020; pp. 480–489.
10. Havasi, M.; Hernández-Lobato, J.M.; Murillo-Fuentes, J.J. Inference in deep Gaussian processes using stochastic gradient Hamiltonian Monte Carlo. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 7506–7516.
11. Duvenaud, D.; Rippel, O.; Adams, R.; Ghahramani, Z. Avoiding pathologies in very deep networks. In Proceedings of the Artificial Intelligence and Statistics, Reykjavik, Iceland, 22–25 April 2014; pp. 202–210.
12. Dunlop, M.M.; Girolami, M.A.; Stuart, A.M.; Teckentrup, A.L. How deep are deep Gaussian processes? *J. Mach. Learn. Res.* **2018**, *19*, 2100–2145.
13. Tong, A.; Choi, J. Characterizing Deep Gaussian Processes via Nonlinear Recurrence Systems. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021; Volume 35, pp. 9915–9922.
14. Agrawal, D.; Papamarkou, T.; Hinkle, J.D. Wide Neural Networks with Bottlenecks are Deep Gaussian Processes. *J. Mach. Learn. Res.* **2020**, *21*, 175–1.
15. Pleiss, G.; Cunningham, J.P. The Limitations of Large Width in Neural Networks: A Deep Gaussian Process Perspective. *arXiv* **2021**, arXiv:2106.06529.
16. Wilson, A.G.; Hu, Z.; Salakhutdinov, R.; Xing, E.P. Deep kernel learning. In Proceedings of the Artificial Intelligence and Statistics, Cadiz, Spain, 9–11 May 2016; pp. 370–378.
17. Salakhutdinov, R.; Hinton, G.E. Using Deep Belief Nets to Learn Covariance Kernels for Gaussian Processes. *Citeseer* **2007**, *7*, 1249–1256.
18. Calandra, R.; Peters, J.; Rasmussen, C.E.; Deisenroth, M.P. Manifold Gaussian processes for regression. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 3338–3345.

19. Ober, S.W.; Rasmussen, C.E.; van der Wilk, M. The promises and pitfalls of deep kernel learning. *arXiv* **2021**, arXiv:2102.12108.
20. Titsias, M. Variational learning of inducing variables in sparse Gaussian processes. In Proceedings of the Artificial Intelligence and Statistics, Clearwater Beach, FL, USA, 16–18 April 2009; pp. 567–574.
21. Titsias, M.; Lawrence, N. Bayesian Gaussian process latent variable model. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 844–851.
22. Matthews, A.G.d.G.; Hensman, J.; Turner, R.; Ghahramani, Z. On sparse variational methods and the Kullback–Leibler divergence between stochastic processes. In Proceedings of the Artificial Intelligence and Statistics, Cadiz, Spain, 9–11 May 2016; pp. 231–239.
23. Lu, C.K.; Yang, S.C.H.; Hao, X.; Shafto, P. Interpretable deep Gaussian processes with moments. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Virtual, 26–28 August 2020; pp. 613–623.
24. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; MIT Press: Cambridge, MA, USA, 2012.
25. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
26. Garnelo, M.; Rosenbaum, D.; Maddison, C.; Ramalho, T.; Saxton, D.; Shanahan, M.; Teh, Y.W.; Rezende, D.; Eslami, S.A. Conditional Neural Processes. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1704–1713.
27. Snelson, E.; Ghahramani, Z.; Rasmussen, C.E. Warped gaussian processes. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 13–18 December 2004; pp. 337–344.
28. Lázaro-Gredilla, M. Bayesian warped Gaussian processes. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1619–1627.
29. Ma, C.; Li, Y.; Hernández-Lobato, J.M. Variational implicit processes. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 4222–4233.
30. Ustyuzhaninov, I.; Kazlauskaite, I.; Ek, C.H.; Campbell, N. Monotonic gaussian process flows. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Sicily, Italy, 26–28 August 2020; pp. 3057–3067.
31. Telgarsky, M. Benefits of depth in neural networks. In Proceedings of the Conference on Learning Theory, New York, NY, USA, 23–26 June 2016; pp. 1517–1539.
32. Pearce, T.; Tsuchida, R.; Zaki, M.; Brintrup, A.; Neely, A. Expressive priors in bayesian neural networks: Kernel combinations and periodic functions. In Proceedings of the Uncertainty in Artificial Intelligence, Virtual, 3–6 August 2020; pp. 134–144.
33. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 1050–1059.
34. Cutajar, K.; Bonilla, E.V.; Michiardi, P.; Filippone, M. andom feature expansions for deep Gaussian processes. In Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, Sydney, Australia, 6–11 August 2017; pp. 884–893.
35. Rahimi, A.; Recht, B. Random features for large-scale kernel machines. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–11 December 2008; pp. 1177–1184.
36. Cho, Y.; Saul, L.K. Kernel methods for deep learning. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 7–10 December 2009; pp. 342–350.
37. Schölkopf, B.; Smola, A.; Müller, K.R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **1998**, *10*, 1299–1319.
38. Girard, A.; Rasmussen, C.E.; Candela, J.Q.; Murray-Smith, R. Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–13 December 2003; pp. 545–552.
39. Muandet, K.; Fukumizu, K.; Dinuzzo, F.; Schölkopf, B. Learning from distributions via support measure machines. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 10–18.
40. Li, Y.; Swersky, K.; Zemel, R. Generative moment matching networks. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 1718–1727.
41. Quiñonero-Candela, J.; Rasmussen, C.E. A unifying view of sparse approximate Gaussian process regression. *J. Mach. Learn. Res.* **2005**, *6*, 1939–1959.
42. Shi, J.; Titsias, M.K.; Mnih, A. Sparse Orthogonal Variational Inference for Gaussian Processes. *arXiv* **2019**, arXiv:1910.10596.
43. Lázaro-Gredilla, M.; Figueiras-Vidal, A.R. Inter-domain Gaussian Processes for Sparse Inference using Inducing Features. *Citeseer* **2009**, *22*, 1087–1095.
44. Dutordoir, V.; Durrande, N.; Hensman, J. Sparse Gaussian processes with spherical harmonic features. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 2793–2802.
45. Dutordoir, V.; Hensman, J.; van der Wilk, M.; Ek, C.H.; Ghahramani, Z.; Durrande, N. Deep Neural Networks as Point Estimates for Deep Gaussian Processes. *arXiv* **2021**, arXiv:2105.04504.
46. Rudner, T.G.; Sejdinovic, D.; Gal, Y. Inter-domain deep Gaussian processes. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 8286–8294.
47. Ober, S.W.; Aitchison, L. Global inducing point variational posteriors for bayesian neural networks and deep gaussian processes. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 8248–8259.

48. GPy. GPy: A Gaussian Process Framework in Python. 2012. Available online: http://github.com/SheffieldML/GPy (accessed on 15 October 2021).
49. Lu, C.K.; Shafto, P. Conditional Deep Gaussian Process: Multi-fidelity kernel learning. *arXiv* **2021**, arXiv:2002.02826.
50. Isserlis, L. On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. *Biometrika* **1918**, *12*, 134–139.
51. Vladimirova, M.; Verbeek, J.; Mesejo, P.; Arbel, J. Understanding priors in bayesian neural networks at the unit level. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6458–6467.
52. Yaida, S. Non-Gaussian processes and neural networks at finite widths. In Proceedings of The First Mathematical and Scientific Machine Learning Conference, Virtual, 20–24 July 2020; pp. 165–192.
53. Zavatone-Veth, J.; Pehlevan, C. Exact marginal prior distributions of finite Bayesian neural networks. *arXiv* **2021**, arXiv:2104.11734.
54. Duvenaud, D.; Lloyd, J.; Grosse, R.; Tenenbaum, J.; Zoubin, G. Structure discovery in nonparametric regression through compositional kernel search. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 17–19 June 2013; pp. 1166–1174.
55. Sun, S.; Zhang, G.; Wang, C.; Zeng, W.; Li, J.; Grosse, R. Differentiable compositional kernel learning for Gaussian processes. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 4828–4837.
56. Mhaskar, H.; Liao, Q.; Poggio, T. Learning functions: When is deep better than shallow. *arXiv* **2016**, arXiv:1603.00988.
57. Dutordoir, V.; Salimbeni, H.; Hambro, E.; McLeod, J.; Leibfried, F.; Artemev, A.; van der Wilk, M.; Deisenroth, M.P.; Hensman, J.; John, S. Pflux: A library for Deep Gaussian Processes. *arXiv* **2021**, arXiv:2104.05674.