Fast optimize-and-sample method for differentiable Galerkin approximations of multi-layered Gaussian process priors

Muhammad F. Emzir*, Niki A. Loppi†, Zheng Zhao‡, Syeda S. Hassan§, and Simo Särkkä§
*Control and Instrumentation Engineering Department, KFUPM, Saudi Arabia
†NVIDIA, Finland
‡Department of Information Technology, Uppsala University, Sweden
§Department of Electrical Engineering Automation, Aalto University, Finland

***Abstract*—Multi-layered Gaussian process (field) priors are non-Gaussian priors, which offer a capability to handle Bayesian inference on both smooth and discontinuous functions. Previously, performing Bayesian inference using these priors required the construction of a Markov chain Monte Carlo sampler. To converge to the stationary distribution, this sampling technique is computationally inefficient and hence the utility of the approach has only been demonstrated for small canonical test problems. Furthermore, in numerous Bayesian inference applications, such as Bayesian inverse problems, the uncertainty quantification of the hyper-prior layers is of less interest, since the main concern is to quantify the randomness of the process/field of interest. In this article, we propose an alternative approach, where we optimize the hyper-prior layers, while inference is performed only for the lowest layer. Specifically, we use the Galerkin approximation with automatic differentiation to accelerate optimization. We validate the proposed approach against several existing non-stationary Gaussian process methods and demonstrate that it can significantly decrease the execution time while maintaining comparable accuracy. We also apply the method to an X-ray tomography inverse problem. Due to its improved performance and robustness, this new approach opens up the possibility for applying the multi-layer Gaussian field priors to more complex problems.**

***Index Terms*—Bayesian learning, Gaussian Processes, Markov chain Monte Carlo, inverse problems, Galerkin approximations**

I. Introduction

Multi-layered Gaussian process (or field) priors are flexible non-Gaussian priors for classification, regression, and inverse problems and they have been recently studied in [1]–[5]. These statistical priors are flexible due to their ability to offer both smoothing and edge preservation capabilities, which typically are considered as competing objectives. Previously, discontinuous functions were predominantly modeled by specific non-Gaussian priors, the total variation prior being the most common [6]. However, upon increasing the number of points in the discretization grid, total variation prior is prone to losing edge preservation [6] and introducing patchy artifacts [7].

A multi-layered Gaussian process (GP) prior can be expressed in terms of a set of stochastic partial differential equations, which can be solved either using spatial grid approximations [2], or using Galerkin type of spectral methods [4]. Until now, the use of multi-layered GP has been restricted to small toy problems since the posterior distributions of each conditionally Gaussian field are obtained via computationally expensive Markov chain Monte Carlo (MCMC) approximations. For example, consider the Bayesian inversion problem in [2], [4], where the authors used the preconditioned Crank–Nicolson (pCN) [8] sampler to draw samples from the resulted posterior distribution. Although the pCN algorithm is simple and has nice analytical properties in very high dimensions, it is infamously known for its strong sample autocorrelation [9]. In addition to the MCMC techniques, other lines of research in multi-layered Gaussian processes inference have been proposed to alleviate the computational burden. These include inducing point and variational methods [1], [10]. In [5] it is shown that one can also use state-space methods (i.e., non-linear stochastic filters and smoothers) to solve the multi-layered GP regression problems. However, this method applies to temporal data only, since their model formulation is restricted to finite-dimensional stochastic differential equations.

In many inference applications, such as Bayesian inverse problems, often the main interest is to recover the posterior distribution of the unknown field from a set of observations. In such cases, drawing samples from all hyper-prior layers is less important. Therefore, we propose an alternative approach, where we optimize the hyper-prior layers, while sampling only the last layer.

Regarding hyper-prior optimization, the negative log-likelihood of the hyper-prior of multi-layered Gaussian process is non-convex due to the log-determinant terms, which are concave functions of a positive definite matrix [11, Theorem 11.25] (see Section II). Therefore, the convexity of the hyper-prior negative log-likelihood with respect to the Galerkin expansion of coefficients is also not guaranteed. The absence of the convexity opens up the possibility of existence of multiple local minima, making global optimization difficult and computationally demanding [12]. Similar non-convexity condition also holds if we use a spatial grid discretization instead of the Galerkin expansion [2]. Moreover, obtaining the Jacobian of the negative log-likelihood analytically becomes challenging due to the presence of the chain of operations, and the involvement of inverse Fourier–Fourier transform pairs. While we could still approximate the Jacobian via finite differentiation, generally, it is numerically unstable and computationally expensive. This has rendered the optimization approach to be less attractive, as the cost of evaluating the optimizer to maintain the quality of the Jacobian could be almost comparable to the time obtaining the MCMC samples [2], [13]. Fortunately, one can obtain an exact Jacobian and other derivatives (up to machine precision) by using auto differentiation approaches [14]. In this work, we use the JAX library [15] to develop an algorithm of similar utility to [4] that builds upon auto differentiation to improve performance.

The contributions of this article are as follows. We propose a Bayesian inference procedure for multi-layered Gaussian process priors, where we perform the optimization of the Galerkin coefficients of hyper-prior layers via auto-differentiation. The posterior distribution of the last layer is calculated via the standard Gaussian form conditioned on the optimized hyper-prior layers. Therefore, our approach can be considered as a hybrid that mixes hyper-prior optimization with standard Gaussian inference. We validate our method with other state-of-the-art non-stationary Gaussian process methods. We also demonstrate that our proposed method can significantly decrease the execution time without compromising the accuracy. In particular, we reduce the execution time of an X-ray tomography Bayesian inverse problem in [4] from 12 days to just around 40 minutes, while achieving better peak-signal-to-noise ratio (PSNR) scores. Moreover, the runtime speed-up is not the only advantage that the proposed optimize-and-sample method has over the work of [4]. Due to the optimization approach, we can also automatically find values for the Matérn covariance function scale parameter parameter, which previously had to be set by the user with educated guesses.

The article is structured as follows. In Section II, we briefly review the multi-layered GP prior formulation. We also show that there is an equivalent neural network realization of this prior. In Section III, we describe our optimization approach to obtain the Galerkin coefficients of the hyper-prior layer via auto-differentiation. In Section IV, we validate the method using two numerical experiments: a one-dimensional denoising problem and an X-ray tomography problem. Conclusions are drawn in Section V. All codes related to the simulations are available from https://anonymous.4open.science/r/JaxDeepSPDE-DEC7.

II. MULTI-LAYERED GAUSSIAN PROCESS PRIOR

In this section, we briefly summarize the multi-layered Gaussian process (or field) prior proposed in [2], [4]. Suppose we would like to infer an unknown process $v : \Omega \to \mathbb{R}, \Omega \subseteq \mathbb{R}^d$ from a set of discrete measurements $y \in \mathbb{R}^m$. In a multi-layered Gaussian process prior with J hyper-layers, the unknown process v is assumed to be the last layer from a collection of $J+1$ layers of conditionally Gaussian random processes, where the first layer is assumed to be a stationary Gaussian process with chosen hyper-prior parameters. Using a Matérn covariance function for the first layer u_0, we can represent the prior as a stochastic partial differential equation (SPDE) of the form

$$\left(1-\ell_0^2\Delta\right)^{\alpha/2} u_0(x) = \sqrt{\beta_0\ell^d}\, w_0(x), \tag{1}$$

where Δ is the Laplacian, $\alpha = \nu + d/2$, ν is a smoothness parameter, $w_0(x)$ is white noise field on $\mathbb{R}^d$, ℓ_0 is the length-scale constant of the Matérn covariance, and $\beta_0 = \sigma_0^2 2^d \pi^{d/2}\Gamma(\alpha)/\Gamma(\nu)$ with σ_0^2 being a scale parameter. In this study, we restrict to $\alpha = 2$ so that (1) becomes second order linear SPDE. For the remaining fields u_j for $j = 1, 2, \dots, J$, we modify the previous SPDE such that the length-scale ℓ_j is modeled as a function of process u_{j-1} with a Matérn covariance function. We select $\kappa_j(x) := 1/\ell_j(x) = g(u_{j-1}(x))$, where g is a smooth positive function $g : \mathbb{R} \to \mathbb{R}_+$. Since the length-scale is always greater than zero, with probability of one, we obtain

$$\left(\kappa(u_{j-1}(x))^2 - \Delta\right) u_j(x) = \sqrt{\beta_j}\kappa(u_{j-1}(x))^{\nu} w_j(x). \tag{2}$$

Assuming periodic boundary condition on the domain Ω, we seek an approximate solution to the SPDE using the Galerkin method. With this boundary condition, it is useful to consider that the fields $\{u_j\}_{0\le j\le J}$ as elements of a complex Hilbert space $\mathcal{H} = \mathcal{L}^2(\Omega)$ where we choose the Hilbert space basis to be the Fourier complex basis $\phi_l = \exp\big(i\, c_d\, x^\top \mathbf{k}(l)\big)$, where c_d is a scaling constant related to Fourier transform and $\mathbf{k}(l)$ is a multi-index unique to each l. The weak form of (2) is given by

$$\langle \nabla u_j, \nabla\varphi\rangle + \langle \kappa^2(u_{j-1})u_j, \varphi\rangle = \beta_j^{1/2}\langle w_j \kappa^{\nu}(u_{j-1}), \varphi\rangle \tag{3}$$

where $\langle\cdot,\cdot\rangle$ is the inner product on the Hilbert space $\mathcal{H}$ and φ is a test function. The $j-$th field u_j is expressed as a linear combination of the Fourier basis functions as $u_j = \sum_{l=-\infty}^{\infty} \hat{u}_{j,l}\phi_l$. The finite Galerkin approximation $u_j^N \in \mathcal{H}_N$, where N denotes a user-specified truncation limit for the Fourier series, satisfies (3) for all $\varphi \in \mathcal{H}_N$, spanned by $\{\phi_l\}_{-N\le l\le N}$, which is a subspace of the full Hilbert space $\mathcal{H}$, spanned by $\{\phi_l\}_{-\infty\le l\le\infty}$. In other words, we require discretization error of each term to be orthogonal to the subspace $\mathcal{H}_N$.

The weak form can be written as a series of mappings as

$$L(\hat{u}_{j-1})\hat{u}_j = \hat{w}_j, \tag{4}$$

where $\hat{w}_j \sim \mathcal{N}(0, I), \hat{w}_j \in \mathbb{C}^{2N-1}$ and $L(\hat{u}_{j-1}) \in \mathbb{C}^{2N-1\times 2N-1}$. $L(\hat{u}_{j-1})$ is referred to as the square root of the precision operator that is a function of the previous field $\hat{u}_{j-1}$, and it is given by

$$L(\hat{u}_{j-1}) = \beta_j^{-\frac{1}{2}}\left(M_N(\kappa_j^{d/2}) - M_N(\kappa_j^{-\nu})D\right). \tag{5}$$

Here, $\kappa_j(x) := \kappa(u_{j-1}^N(x))$, and the constant β_j are computed using the same procedure as for the first layer. The matrix $D \in \mathbb{C}^{2N-1\times 2N-1}$ is a diagonal matrix with elements equivalent to the eigenvalues of the Laplacian, following $-\Delta u = \sum_{l=-\infty}^{\infty} \lambda_i \langle u, \phi_i\rangle \phi_i$. The matrix $M_N(z) \in \mathbb{C}^{2N-1\times 2N-1}$ is the matrix representation of the multiplication operator corresponding to a field z. In our case it has the form $M_N(\kappa(u_j)^\gamma)$, and it can be obtained by applying the mapping $\kappa^\gamma(\cdot)$ to the inverse Fourier transform of $\hat{u}_j$, and taking the Fourier transform of the result.

We assume that the measurement is a linear function of the field in the last layer $v = u_J$; that is $y = Hv + e = \hat{H}\hat{u}_J + (Hv - \hat{H}\hat{u}_J) + e, e \sim \mathcal{N}(0, R)$. The matrix $\hat{H} \in \mathbb{C}^{m \times (2N-1)}$ is the spectral representation of the linear measurement operator H, $R \in \mathbb{R}^{m \times m}$ is a positive definite matrix, and the term $(Hv - \hat{H}\hat{u}_J)$ approaches zero (in the Hilbert space norm) as N approaches infinity.

The white noise Galerkin coefficients are sampled as $\bar{w} := \{\hat{w}_j\}_{0 \le j \le J-1}$, instead of $\{\hat{u}_j\}_{0 \le j \le J-1}$. This is referred to as a non-centered formulation and it makes the calculation of the log posterior simpler, since it breaks the dependence of Galerkin coefficients in each layer. For more details, see [4]. The construction of a multi-layered Gaussian process given by (4) is general in a sense that many other multi-layered Gaussian process formulations can be cast into this framework as shown in [3]. Therefore, the use of this approach is not only restricted to inverse problems, but it can be used also in other Gaussian process applications such as regression and classification. Note that to obtain $\hat{u}_{J-1}$, all Galerkin coefficients from the previous layers need to be obtained via (4), therefore, $\hat{u}_{J-1}$ is a function of all white noises Galerkin coefficients $\bar{w}$. Under this formulation, the probability distribution of $\bar{w}$ is given by

$$\mathbb{P}(d\bar{w}|y) \propto \exp\left(-\Psi(\bar{w}, y)\right) \mathbb{P}(d\bar{w}), \tag{6a}$$

$$\Psi(\bar{w}, y) = \frac{1}{2}\|y\|_Q^2 + \frac{1}{2}\log\det(Q), \tag{6b}$$

$$Q = \hat{H}L(\hat{u}_{J-1})^{-1}L(\hat{u}_{J-1})^{-\top}\hat{H}^\top + R. \tag{6c}$$

Once we have computed $\hat{u}_{J-1}$, a standard Gaussian form is used to calculate the mean and covariance matrix of Fourier coefficients of the field of the last layer $u_J = v$ as

$$\hat{v} \sim \mathcal{N}(Z(\hat{u}_{J-1})^{-1}\hat{H}^\top R^{-1} y, Z(\hat{u}_{J-1})), \tag{7}$$

where $Z = (L(\hat{u}_{J-1})^\top L(\hat{u}_{J-1}))^{-1} + \hat{H}^\top R \hat{H}$. Subsequently, since the inverse Fourier transform of the Fourier elements $\hat{v}$ can be expressed as matrix vector multiplication: $v = \Phi\hat{v}$, the stochastic field (or process) v is distributed according to

$$v \sim \mathcal{N}\left(\Phi\mathbb{E}[\hat{v}], \Phi\mathbb{V}[\hat{v}]\Phi^\top\right). \tag{8}$$

Here, $\mathbb{E}$ and $\mathbb{V}$ are the expectation and covariance operators, respectively and $\mathbb{V}[\hat{v}] := \mathbb{E}[\hat{v}\hat{v}^\top] - \mathbb{E}[\hat{v}]\mathbb{E}[\hat{v}]^\top$.

III. Proposed Method

In this section, we present our hybrid Bayesian inference approach. Essentially, we replace the Metropolis sampling part in the Metropolis-within-Gibbs sampling algorithm proposed in [4] with an optimization procedure. For one complete measurement set $y^{(k)}$, we initialize the Galerkin coefficients of each hyper-prior layer and their respective field strength with some initial values $\{\hat{w}_j\}_{0 \le j \le J}, \{\sigma_j\}_{0 \le j \le J}$. Then we run an optimization procedure to obtain the optimized parameters $\{\hat{w}_j\}_{0 \le j \le J}, \{\sigma_j\}_{0 \le j \le J}$. The optimization via auto-differentiation procedure is possible since the function Ψ can be considered differentiable as follows. If we consider the negative log likelihood function Ψ as a function of both $\bar{w}, \bar{w}^*$; i.e., $\Psi : (\bar{w}, \bar{w}^*) \to \mathbb{R}$, then the derivative of Ψ is well defined even though Ψ is non-holomorphic with respect to $\bar{w}$. For an infinitesimal change $\delta z = (\delta\bar{w}, \delta\bar{w}^*)$, the change of Ψ is given by $\delta\Psi = 2\,\mathrm{Re}\left[(\partial_{\bar{w}^*}\Psi)^\top \delta\bar{w}\right]$. By a well-known result from complex analysis [16], the stationary points are points where the gradient $\partial_{\bar{w}^*}\Psi$ vanishes, and the optimum decrease in Ψ happens when $\delta\bar{w}$ is in the direction of $-\partial_{\bar{w}^*}\Psi$.

Using the optimized hyper-prior Galerkin coefficients, we solve the square root of the precision matrix for the last hyper-prior layer via Equations (4) and (5). Then we use (8) to calculate the conditional Gaussian distribution of the last layer. This procedure is summarized in Algorithm 1.

Algorithm 1 Multi-layered GP Bayesian inference with optimization

Precondition: $\left\{y^{(k)}\right\}_{1 \le k \le N_s}$ set of N_s independent measurement samples, J number of layers, $\left\{\hat{w}_j^\bullet\right\}_{0 \le j \le J}$ initial Galerkin coefficients, $\left\{\sigma_j^\bullet\right\}_{0 \le j \le J}$ initial field strength.

```
1: function MGP OPTIMIZE THEN INFERENCE($\left\{y^{(k)}\right\}, \{\hat{w}_{j,0}\}_{0 \le j \le J-1}, \{\sigma_{j,0}\}_{0 \le j \le J-1}$)
2:     for $k \leftarrow 1$ to $N_s$ do
3:         $y \leftarrow y^{(k)}$
4:         $\{\hat{w}_j\}_{0 \le j \le J} \leftarrow \left\{\hat{w}_j^\bullet\right\}_{0 \le j \le J}$
5:         $\{\sigma_j\}_{0 \le j \le J} \leftarrow \left\{\sigma_j^\bullet\right\}_{0 \le j \le J}$
6:         Obtain optimized $\{\sigma_j\}_{0 \le j \le J-1}$,$\{\hat{w}_j\}_{0 \le j \le J-1}$ w.r.t. (6b).
7:         Compute $\hat{u}_{J-1}$ and $L(\hat{u}_{J-1})$ via eq. (4).
8:         Compute (7), store as $\left\{\mathbb{E}[\hat{v}^{(k)}], \mathbb{V}[\hat{v}^{(k)}]\right\}$.
9:         Compute (8), store as $\left\{\mathbb{E}[v^{(k)}], \mathbb{V}[v^{(k)}]\right\}$.
10:
11:         $\ell^{(k)} \leftarrow 1/\kappa(u_{J-1})$
12:     end for
13:     return $\left\{\mathbb{E}[v^{(k)}], \mathbb{V}[v^{(k)}], \ell^{(k)}\right\}_{1 \le k \le N_s}$
14: end function
```

A wide class of optimization procedures can be used to obtain the optimized parameters $\{\sigma_j\}_{0 \le j \le J}$,$\{\hat{w}_j\}_{0 \le j \le J}$ with respect to (6b). In this work, we consider ADAM [17], Rmsprop, and AdaHessian [18] optimizers. While the first two methods are popular within the machine learning community, AdaHessian is a recently published alternative. It was proposed as an approximation of Newton based optimization where the Hessian matrix is approximated as a diagonal matrix and the diagonal elements are computed via Hutchinson's methods [19]. This approach seems to be cheaper to evaluate compared to many Newton/quasi-Newton approaches as the cost of evaluating a diagonalized Hessian and a vector is similar to the ordinary gradient backpropagation.

IV. Numerical Experiments

In this section, we compare the proposed multi-layered Gaussian process prior (MGP) via optimization approach with other methods in the literature. In particular, we compare our method with the deep state space GP (DGP) [5] for a one-dimensional signal denoising problem, and filtered back-projection (FBP) and Tikhonov regularization for an X-ray tomography problem. The comparison of MGP with vanilla GP methods and other non-stationary GP methods is not performed here as they have been studied previously in [5].

A. SIGNAL DENOISING PROBLEM

In this section, we present a numerical example of a signal denoising problem using two artificial signals as follows:

$$v(t) = \begin{cases} \exp\left(4 - \frac{1}{2t-4t^2}\right), & t \in (0, 0.5), \\ 1, & t \in [0.7, 0.8], \\ -1, & t \in (0.8, 0.9], \\ 0, & \text{otherwise.} \end{cases} \tag{9a}$$

$$v(t) = \begin{cases} 1, & t \in \left[\frac{1}{7}, \frac{2}{7}\right), \\ 0.6, & t \in \left[\frac{3}{7}, \frac{4}{7}\right), \\ 0.4, & t \in \left[\frac{5}{7}, \frac{6}{7}\right), \\ 0, & \text{otherwise.} \end{cases} \tag{9b}$$

Notice that the signal $v(t)$ defined in (9a) contains both smooth Gaussian bell in $(0, 0.5)$ and a sudden jump. Denoising this type of signal using total variation prior would normally result in a patchy restoration in $(0, 0.5)$. For the simulations, we take the artificial signal, superpose it with noise $e \sim \mathcal{N}(0, 0.1^2 I)$, measure it at a set of temporal grid points and finally compute the posterior, given the measurements. This process is then repeated for 100 independent noise realizations to gather statistics for error metrics. We use mean absolute error (MAE), root mean squared error (RMSE), negative log predictive distribution (NLPD), and peak signal to noise ratio (PSNR) metrics to compare the different methods, where the error is given by the difference between the estimated mean and the ground truth of the unknown v. Instead of selecting κ to be an exponential function as in [4], we choose $\kappa = \log(\exp(x) + 1)$ for numerical stability. This function also allows an exponential decay of the length scale near the jump points, while satisfying the analytic growth rate conditions required to ensure the Galerkin series convergence. We use 127 Fourier basis functions, that is, $N = 64$, and the number of measurement temporal grid points is set to 255.

From Tables I and II, we can see that the estimation results using MGP with any optimizer outperforms DGP (solved either via the extended Kalman smoother (EKFS) or the cubature Kalman smoother (CKFS)), with the PSNR values of MGP being almost double compared to those of DGP. One exception is the RMSE value of signal (9a), where the DGP performs better.

Fig. 1a shows the mean estimate for the signal (9a). Here, we can observe that the increase in MGP RMSE appears to be caused by the Gibbs phenomena which results in large squared error values near the jump points. This does not occur in DGP since no Fourier transformation is involved. In the bell-shaped region, AdaHessian outperforms ADAM and Rmsprop by yielding a smoother estimate. Fig. 1c plots the optimized length scale associated with the mean estimate in Fig. 1a. The length scale obtained with AdaHessian exhibits smaller variation relative to ADAM and Rmsprop. The length scales obtained with ADAM and Rmsprop optimizations have more pronounced local maxima and minima which results in mean estimates v that resemble straight lines joined together, which is an indication of over-smoothing.

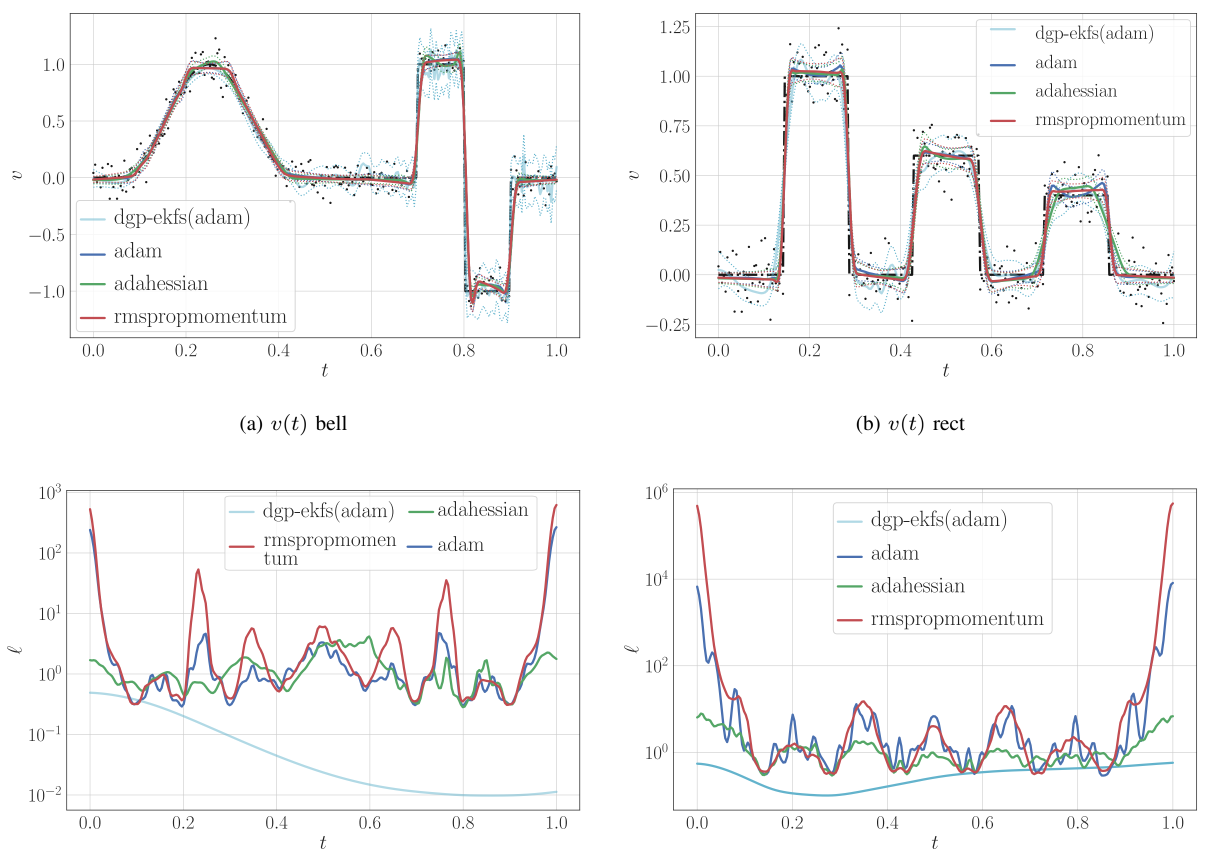
Fig. 1b plots the mean estimate for the signal (9b). Here, the MGP estimate using ADAM optimization has the best performance in terms of both RMSE, MAE and PSNR, although, the MGP estimate using Rmsprop appears visually closer to the ground truth signal. Fig. 1d plots the optimized length scale associated with the mean estimate in Fig. 1b. Again, the length scale obtained with AdaHessian exhibits smaller variation relative to Adam and Rmsprop. In this case, the AdaHessian produces a mean estimate that under-fits the last rectangular section.

B. X-RAY TOMOGRAPHY

In this section we apply the proposed method to a simulated X-ray tomography problem using the Shepp–Logan phantom, where the unknown is a collection of ellipses in a two dimensional domain. The elements of the measurement matrix H correspond to the Radon operator that is given by applying the Radon transform to the basis functions [4]. For this numerical simulation, we use 63 Fourier basis functions for each dimension. For the measurements, we take 45 projective measurements out of 180 using parallel beams, with the height of the sinogram equals to 255 pixels. The measurement noise standard deviation is set to 0.01. Subsequently, we upscale the image during the inverse Fourier step so that the unknown has a resolution 1023×1023 pixels. For simplicity, we set $N_s = 1$.

The noise and reconstruction artifacts in this problem are clearly noticeable. This can be seen from the filtered back-projection (FBP) reconstruction where considerable amounts of noise and artifacts appear; see Fig. 2f for the FBP reconstruction. As a reference, we also use Tikhonov regularization to obtain the Galerkin coefficients of the last layer where the regularization constant is determined by grid search optimization. The reconstructed image is given by Fig. 2e. The reconstructed images using MGP via different optimizers are shown in Fig. 2b, 2d, 2c for the Rmsprop, ADAM, and AdaHessian respectively, while their length-scales are shown in Figures 2g, 2i, and 2h respectively. Table III shows the error metrics for the X-ray tomography case. We can see that the MGP reconstructions have significantly lower MAE and RMSE values as well as better PSNR scores relative to FBP. MGP reconstructions also outperform Tikhonov regularization in terms of MAE and PNRS, while having similar RMSE.

To give context to the efficiency of the proposed approach, we also benchmarked the performance of the X-ray tomography code associated with [4]. It uses the same test case with a configuration analogous to ours, apart from the Phantom scale being $[0, 1]$ instead of $[0, 2]$. To achieve the reported PSNR value of 22.246 in [4], approximately a million samples needs to be taken which equates to a total execution time of 12.59



(a) $v(t)$ bell

(b) $v(t)$ rect

(c) $\ell(t)$ -bell

(d) $\ell(t)$ -rect

Fig. 1: Realization no 49 of the estimation results for signals described in (9a) and (9b) respectively. In 1a and 1b, the ground truth is the black dash-dotted line, the mean estimates are the solid lines, and the two standard deviation confidence interval are given by the dotted lines of each color.

days. The total execution time to obtain the results in Table III is 71 minutes with AdaHessian and 41 minutes with Rmsprop and ADAM using a single NVIDIA Tesla V100 GPU. This gives confidence that an optimizer-based solution approach can significantly decrease the execution time of Gaussian processes formulated with Galerkin expansion.

V. Conclusions and Future Outlook

In this article, we described a Bayesian inference procedure for multi-layered Gaussian process priors where the hyper-prior layers are obtained via optimization. We showed that the hyper-prior layer parameter results via optimization with auto differentiation can lead to an equivalent if not better estimate of the last layer compared to MCMC procedure described in [4]. Compared to the DGP method of [5], the estimation results of one dimensional signal denoising are substantially better in many metrics and also closely resemble the ground truth. For the tomography problem, the proposed Bayesian inference resulted in a reconstructed image that has lower MAE and higher PSNR scores relative to those obtained with FBP and Tikhonov methods.

References

[1] A. Damianou and N. D. Lawrence, "Deep Gaussian processes," in *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, vol. 31. PMLR, 29 Apr–01 May 2013, pp. 207–215.

[2] L. Roininen, M. Girolami, S. Lasanen, and M. Markkanen, "Hyperpriors for Matérn fields with applications in Bayesian inversion," *Inverse Problems & Imaging*, vol. 13, no. 1, pp. 1–29, 2019.

[3] M. M. Dunlop, M. A. Girolami, A. M. Stuart, and A. L. Teckentrup, "How deep are deep Gaussian processes?" *Journal of Machine Learning Research*, vol. 19, no. 54, pp. 1–46, 2018.

[4] M. F. Emzir, S. J. Lasanen, Z. Purisha, L. Roininen, and S. Särkkä, "Non-stationary multi-layered Gaussian priors for Bayesian inversion," *Inverse Problems*, vol. 37, no. 1, Dec. 2020.

[5] Z. Zhao, M. Emzir, and S. Särkkä, "Deep state-space Gaussian processes," *Statistics and Computing*, vol. 31, no. 6, p. 75, 2021.

[6] M. Lassas and S. Siltanen, "Can one use total variation prior for edge-preserving Bayesian inversion?" *Inverse Problems*, vol. 20, no. 5, pp. 1537–1563, Aug. 2004.

[7] J. Tang, B. E. Nett, and G.-H. Chen, "Performance comparison between total variation (TV)-based compressed sensing and statistical iterative reconstruction algorithms," *Physics in Medicine and Biology*, vol. 54, no. 19, pp. 5781–5804, sep 2009.

[8] S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White, "MCMC methods for functions: Modifying old algorithms to make them faster," *Statistical Science*, vol. 28, no. 3, pp. 424–446, 2013.

TABLE I: Metrics comparison of signal (9a) denoising.

Methods	MAE↓	RMSE↓	PSNR↑	NLPD ↓
MGP				
AdaHessian	$\mathbf{0.037 \pm 0.004}$	0.099 ± 0.004	$\mathbf{20.351 \pm 0.426}$	-239.329 ± 1.288
ADAM	0.041 ± 0.004	0.110 ± 0.005	20.155 ± 0.523	-226.334 ± 2.331
Rmsprop	0.040 ± 0.003	0.110 ± 0.005	20.056 ± 0.555	$\mathbf{-241.906 \pm 1.689}$
DGP-ekfs				
ADAM	0.044 ± 0.003	0.072 ± 0.008	12.700 ± 0.683	-169.891 ± 9.542
BFGS	0.044 ± 0.003	$\mathbf{0.067 \pm 0.005}$	13.043 ± 0.528	-169.488 ± 6.129
Rmsprop	0.045 ± 0.003	0.072 ± 0.009	12.656 ± 0.680	-165.362 ± 9.898
DGP-ckfs				
ADAM	0.044 ± 0.003	0.075 ± 0.003	12.352 ± 0.460	-169.355 ± 5.905
BFGS	0.043 ± 0.003	0.074 ± 0.005	12.425 ± 0.431	-169.598 ± 5.464
Rmsprop	0.044 ± 0.003	0.075 ± 0.005	12.358 ± 0.466	-169.338 ± 5.911

TABLE II: Metrics comparison of signal (9b) denoising.

Methods	MAE↓	RMSE↓	PSNR↑	NLPD ↓
MGP				
AdaHessian	0.040 ± 0.004	0.074 ± 0.004	23.472 ± 0.598	-242.618 ± 1.356
ADAM	0.035 ± 0.004	0.069 ± 0.004	$\mathbf{24.143 \pm 0.691}$	$\mathbf{-244.643 \pm 2.760}$
Rmsprop	$\mathbf{0.034 \pm 0.004}$	$\mathbf{0.069 \pm 0.003}$	23.811 ± 0.531	-226.817 ± 1.967
DGP-ekfs				
ADAM	0.055 ± 0.003	0.092 ± 0.003	11.012 ± 0.288	-134.403 ± 7.286
BFGS	0.056 ± 0.003	0.093 ± 0.003	10.956 ± 0.283	-132.041 ± 7.638
Rmsprop	0.054 ± 0.003	0.090 ± 0.003	11.138 ± 0.314	-140.291 ± 6.777
DGP-ckfs				
ADAM	0.052 ± 0.003	0.086 ± 0.003	11.245 ± 0.296	-146.762 ± 6.145
BFGS	0.052 ± 0.003	0.087 ± 0.003	11.194 ± 0.292	-145.279 ± 6.284
Rmsprop	0.051 ± 0.003	0.085 ± 0.003	11.323 ± 0.307	-148.984 ± 5.936

TABLE III: Metrics comparison of the two dimensional X-ray tomography case.

Methods	MAE ↓	RMSE ↓	PSNR ↑
Tikhonov	0.076	0.158	22.946
FBP	0.183	0.265	20.729
MGP			
AdaHess	0.071	0.166	23.145
ADAM	**0.070**	**0.158**	**23.428**
Rmsprop	**0.070**	0.160	23.277

[9] K. J. H. Law, "Proposals which speed up function-space MCMC," *Journal of Computational and Applied Mathematics*, vol. 262, pp. 127–138, 2014.

[10] H. Salimbeni and M. Deisenroth, "Doubly stochastic variational inference for deep Gaussian processes," *Advances in neural information processing systems*, vol. 30, 2017.

[11] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley & Sons, 2019.

[12] Y.-A. Ma, Y. Chen, C. Jin, N. Flammarion, and M. I. Jordan, "Sampling can be faster than optimization," *Proceedings of the National Academy of Sciences*, vol. 116, no. 42, pp. 20 881–20 885, Sep 2019.

[13] F. Lindgren, H. Rue, and J. Lindström, "An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach," *J. Royal Stat. Soc. B*, vol. 73, no. 4, pp. 423–498, 2011.

[14] A. Griewank and A. Walther, *Evaluating Derivatives : Principles and Techniques of Algorithmic Differentiation, Second Edition*. Society for Industrial and Applied Mathematics, Jan 2008.

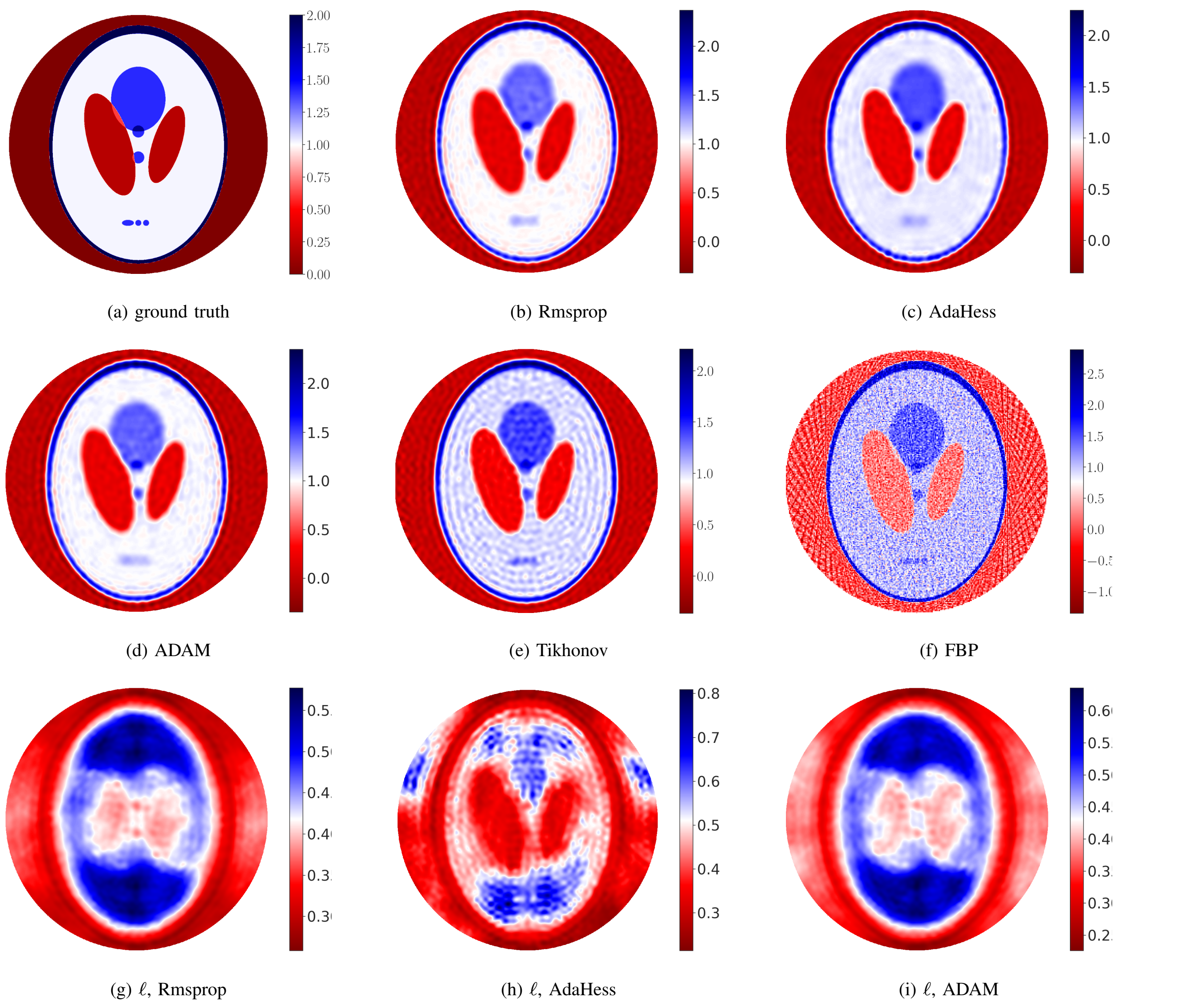
[15] R. Frostig, M. J. Johnson, and C. Leary, "Compiling machine learning programs via high-level tracing," in *2018 1st Conference on Systems for Machine Learning (SysML)*, Dec. 2018.

[16] Z. Nehari, *Introduction to Complex Analysis*, ser. College mathematics series. Allyn & Bacon, 1968.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations (ICLR), San Diego.*, Dec 2015.

[18] Z. Yao, A. Gholami, S. Shen, M. Mustafa, K. Keutzer, and M. W. Mahoney, "ADAHESSIAN: An adaptive second order optimizer for machine learning," *arXiv Preprint arXiv:2006.00719*, Jun 2020.

[19] M. Hutchinson, "A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines," *Communications in Statistics - Simulation and Computation*, vol. 19, no. 2, pp. 433–450, jan 1990.



(a) ground truth (b) Rmsprop (c) AdaHess

(d) ADAM (e) Tikhonov (f) FBP

(g) ℓ, Rmsprop (h) ℓ, AdaHess (i) ℓ, ADAM

Fig. 2: Comparison of reconstruction results using MGP with Rmsprop, AdaHessian, and ADAM, their length respective scales, and the reconstruction result using Tikhonov regularization ($\lambda = 7.50 \times 10^{-3} \|H/\sigma_r\|$, where σ_r is the measurement noise standard deviation, $\|\cdot\|$ is matrix norm) and filtered back projection.