Malleable Kernel Interpolation for Scalable Structured Gaussian Process

Ban, Hanyuan; Riemens, Ellen H.J.; Rajan, Raj Thilak

DOI
10.23919/EUSIPCO63174.2024.10715101

Publication date
2024

Document Version
Final published version

Published in
32nd European Signal Processing Conference, EUSIPCO 2024 - Proceedings

Malleable Kernel Interpolation for Scalable Structured Gaussian Process

Hanyuan Ban, Ellen H. J. Riemens, Raj Thilak Rajan
Signal Processing Systems, *Faculty of EEMCS, Delft University of Technology*, Delft, The Netherlands

***Abstract*—Gaussian process regression (GPR), is a powerful non-parametric approach for data modeling, which has garnered considerable interest in the past decade, however its widespread application is impeded by the significant computational burden for larger datasets. The computational complexity for both inference and hyperparameter learning in GPs lead to $\mathcal{O}(N^3)$ for N training points. The current state-of-the-art approximations, such as structured kernel interpolation (SKI)-based methods e.g., Kernel Interpolation for Scalable Structured Gaussian Process (KISSGP), have emerged to mitigate this challenge by providing a scalable inducing point alternatives. However, the choice of the optimal number of grid points, which influences the accuracy and efficiency of the model, typically remains fixed and is chosen arbitrarily. In this work, we introduce a novel approximation framework, Malleable KISSGP (MKISSGP), which dynamically adjusts grid points using a new hyperparameter of the model called *density*, which adapts to the changes in the kernel hyperparameters in each training iteration. In comparison with the state-of-the-art KISSGP and irrespective of changes in hyperparameters, our proposed MKISSGP algorithm exhibits consistent error levels in the reconstruction of the kernel matrix, and offers reduced computational complexity. We present extensive simulations to validate the improved performance of the proposed MKISSGP, and give directions for future research.**

***Index Terms*—Gaussian process regression, Low-rank approximation, Structured kernel interpolation, KISSGP**

I. Introduction

Gaussian process regression (GPR) is a non-parametric Bayesian regression technique used for modeling (nonlinear) relationships, and provides a principled way to quantify uncertainty in predictions, which is crucial in decision-making and risk assessment [1], [2]. In the past decades, GPR has gained increasing attention in addressing challenges in diverse fields e.g., in predicting atomistic properties in Chemistry [3][4], in nonlinear model predictive control in the domain of control theory [5][6] and localization in sensor network [7]. More recently, GPR networks have been proposed to combine the structural properties of Bayesian neural networks with the non-parametric flexibility of Gaussian Process [8], and other extensions such as Deep Gaussian Processes have been explored [9].

Despite the numerous advantages, the underlying optimization of GPR implicitly requires the inversion of a kernel matrix, which costs $\mathcal{O}(N^3)$, where N is the number of data points [1]. In many real-world applications, either the vast quantity of training data N, or the requirement of frequent updates prevents the direct use of GPR for large datasets. To alleviate this high computational complexity, the GPR cost function is typically distributed among various nodes [10], [11], [12], or various local and global approximations have been proposed [13][14][15][16]. There are numerous approximation methods, such as Nyström approximation [17], fully independent training conditional (FITC) [18] and sparse spectral Gaussian process (SSGP) [19], which achieve a good balance between accuracy and time. However, the current state-of-the-art low-rank approximation based on the structural kernel interpolation (SKI) framework is kernel interpolation for scalable structured Gaussian process (KISSGP) [20], which leverages a pre-selected set of grid points to interpolate the kernel matrix. KISSGP achieves a time complexity of $\mathcal{O}(N+M^2)$ where $M \ll N$ is the number of chosen grid points, and can be further reduced to $\mathcal{O}(N + M \log M)$ when the Toeplitz structure of the kernel is exploited [15][20]. However, in current SKI methods, there is a lack of well-defined strategy for determining the optimal value of M. Notably, while literature offers methods to address the exponential growth of M [16][21][22], the precise determination of the number of grid points remains unclear.

In this paper, we present a novel low-rank approximation framework, denoted as malleable kernel interpolation for scalable structured Gaussian process (MKISSGP), which extends the capabilities of the well-established state-of-the-art SKI-based KISSGP approximation, with the determination of flexible grid points. The determination of the number of grid points M_{opt}, is our key contribution in this work, leading to a reduced computational complexity of $\mathcal{O}(N + M_{opt}^2)$, where M_{opt} is the optimal number of grid points to reach a desired accuracy. This avoids the arbitrary choice of M which may lead to insufficient accuracy or excessive computational complexity.

II. Gaussian Process Regression

Consider a regression model $y = f(\mathbf{x}) + w$ relating a $D-$dimensional input $\mathbf{x}$ to an output y under Gaussian noise $w \sim \mathcal{N}(0, \sigma_n^2)$ assumption. The underlying function can be modeled as a Gaussian Process (GP) i.e., $f(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, k(\mathbf{x}, \mathbf{x}'))$, where without loss of generality we assume zero-mean, and introduce the scalar kernel function $k(\cdot,\cdot)$ relating two input vectors [1]. Now, consider a dataset of N inputs $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ with corresponding outputs $\mathbf{y} = \{y_n\}_{n=1}^N$, then the values of the function at a finite set of inputs are jointly Gaussian i.e.,

$$\mathbf{f} = [f(\mathbf{x_1}), \ldots, f(\mathbf{x_N})] \sim \mathcal{N}(\mathbf{0}, \mathbf{K}), \tag{1}$$

where $\mathbf{K}$ is a $N \times N$ kernel matrix whose element-wise entries $[\mathbf{K}]_{ij} = k(\mathbf{x_i}, \mathbf{x_j})$ form the scalar kernel function. In this work, we use the radial basis function (RBF) kernel i.e.,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_s^2 \exp\left[-\frac{1}{2l^2}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \right], \tag{2}$$

which contain the hyperparameters $\boldsymbol{\theta} = [\sigma_s^2,\ l]^T$ i.e., the signal power (σ_s^2) and length-scale (l) respectively. Given the input-output pairs $\{\mathbf{X}, \mathbf{y}\}$, the hyperparameters $\boldsymbol{\theta}$ and the noise power σ_n, the predictive distribution of the test inputs $\mathbf{X}_*$ is a joint Gaussian distribution conditioned on the given information i.e.,

$$\mathbf{f}_* | \mathbf{X}_*, \mathbf{X}, \mathbf{f}, \boldsymbol{\theta}, \sigma_n \sim \mathcal{N}(\bar{\mathbf{f}}_*, \mathrm{cov}(\mathbf{f}_*)), \tag{3}$$

where

$$\bar{\mathbf{f}}_* = \mathbf{K}_{\mathbf{X},*}^T (\mathbf{K}_{\mathbf{X},\mathbf{X}} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}, \tag{4a}$$

$$\mathrm{cov}(\mathbf{f}_*) = \mathbf{K}_{*,*} - \mathbf{K}_{\mathbf{X},*}^T (\mathbf{K}_{\mathbf{X},\mathbf{X}} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{K}_{\mathbf{X},*}. \tag{4b}$$

Here, $\mathbf{I}$ denotes the identity matrix, and $\mathbf{K}_{\mathbf{X},\mathbf{X}}, \mathbf{K}_{\mathbf{X},*}$ and $\mathbf{K}_{*,*}$ are the kernel matrices between $\mathbf{X}$ and $\mathbf{X}$, $\mathbf{X}$ and $\mathbf{X}_*$, and $\mathbf{X}_*$ and $\mathbf{X}_*$, respectively. Let $\det(\cdot)$ denote the determinant of a matrix, then the marginal likelihood of the data, conditioned only on the hyperparameters $\boldsymbol{\theta}$ is analytically

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) \propto 0.5[&\mathbf{y}^T (\mathbf{K}_{\mathbf{X},\mathbf{X}} + \sigma_n^2)^{-1} \mathbf{y} \\ &+ \log\det\big(\mathbf{K}_{\mathbf{X},\mathbf{X}} + \sigma_n^2\big) + N \log 2\pi]. \end{aligned} \tag{5}$$

Observe that, the computational bottleneck of solving for $\log\det\big(\mathbf{K}_{\mathbf{X},\mathbf{X}} + \sigma_n^2\big)$ and $(\mathbf{K}_{\mathbf{X},\mathbf{X}} + \sigma_n^2)^{-1}\mathbf{y}$ typically requires $\mathcal{O}(N^3)$ time complexity using the Cholesky decomposition of the kernel, which limits the use of GPR for large datasets.

A. *KISSGP*

Conventionally, approximation methods reduce the time-complexity of GPR by reducing the rank of the kernel matrix [13]. For instance, let $\mathbf{U} = \{\mathbf{u}_m\}_{m=1}^M$ be a set of M predefined inducing points such that $M \ll N$, then subset of regressors (SoR) method [23] uses the following approximation

$$\mathbf{K}_{\mathbf{X},\mathbf{X}} \approx \tilde{\mathbf{K}} = \mathbf{K}_{\mathbf{X},\mathbf{U}} \mathbf{K}_{\mathbf{U},\mathbf{U}}^{\mathbf{-1}} \mathbf{K}_{\mathbf{U},\mathbf{X}}, \tag{6}$$

where $\mathbf{K}_{\mathbf{X},\mathbf{U}}$ represents the kernel evaluated at the corresponding training and inducing inputs, while $\mathbf{K}_{\mathbf{U},\mathbf{U}}$ is the kernel evaluated only from the respective inducing inputs. KISSGP further reduces the computational complexity by finding a suitable approximation for $\mathbf{K}_{\mathbf{X},\mathbf{U}}$ [20]. For any input $\mathbf{x}_i$, $1 \leq i \leq N$, a set of M weights $\{w_{i,m}\}_{m=1}^M$ corresponding to all grid points $\mathbf{U}$ is found, such that

$$k(\mathbf{x}_i, \mathbf{u}_j) \approx \sum_{m=1}^{M} w_{i,m} k(\mathbf{u}_m, \mathbf{u}_j), \tag{7}$$

where $k(\mathbf{x}_i, \mathbf{u}_j) = [\mathbf{K}_{\mathbf{X},\mathbf{U}}]_{i,j}$ and $k(\mathbf{u}_m, \mathbf{u}_j) = [\mathbf{K}_{\mathbf{U},\mathbf{U}}]_{m,j}$. Let all the weights be populated in a $N \times M$ interpolation matrix $\mathbf{W}$, such that $\mathbf{K}_{\mathbf{X},\mathbf{U}} \approx \mathbf{W}\mathbf{K}_{\mathbf{U},\mathbf{U}}$, where $\mathbf{W}$ could be very sparse. Substituting this expression in (6), we have

$$\mathbf{K}_{\mathbf{X},\mathbf{X}} \approx \tilde{\mathbf{K}} = \mathbf{W}\mathbf{K}_{\mathbf{U},\mathbf{U}}\mathbf{W}^T, \tag{8}$$

which is the Scalable Kernel Interpolation (SKI) framework or KISSGP. Now, substituting (8) in (5), the modified log likelihood is

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{U}, \boldsymbol{\theta}) \approx 0.5[&\mathbf{y}^T (\tilde{\mathbf{K}} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} \\ &+ \log\det\Big(\tilde{\mathbf{K}} + \sigma_n^2 \mathbf{I}\Big) + N \log 2\pi]. \end{aligned} \tag{9}$$

Note that $\mathbf{K}_{\mathbf{X},\mathbf{X}}$ has been replaced by $\tilde{\mathbf{K}}$, which consists of sparse $\mathbf{W}$, and $\mathbf{K}_{\mathbf{U},\mathbf{U}}$ with a Toeplitz structure due to the equispaced gridpoints. Exploiting this structure of $\tilde{\mathbf{K}}$ reduces the time complexity for the calculations of both $(\tilde{\mathbf{K}} + \sigma_n^2 \mathbf{I})^{-1}\mathbf{y}$ and $\log\det\Big(\tilde{\mathbf{K}} + \sigma_n^2 \mathbf{I}\Big)$, and facilitates a reduction in time complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(N + M^2)$. There are numerous approaches to construct the sparse interpolation matrix $\mathbf{W}$, for e.g., cubic convolution interpolation (CCI), which was implemented in the original KISSGP work [20], which results in 4^D non-zero entries per row [24].

III. Proposed Low-Rank Approximation

One of the key features of KISSGP is the number of fixed gridpoints M in the learning phase, which is typically arbitrarily chosen. Furthermore, for accurate kernel reconstruction, a higher M is required, which in turn leads to a computational bottleneck for larger datasets. To overcome this challenge, we make the observations that, for RBF kernels M depends on the length scale l to achieve a desired accuracy and for all interpolation methods the distance between grid points is naturally fixed. In pursuit of a unified metric, we combine these observations and define a novel *density* metric ρ, which relates the distance between successive grid points $d = \|\mathbf{u}_{m+1} - \mathbf{u}_m\|_2$ and the kernel length l as

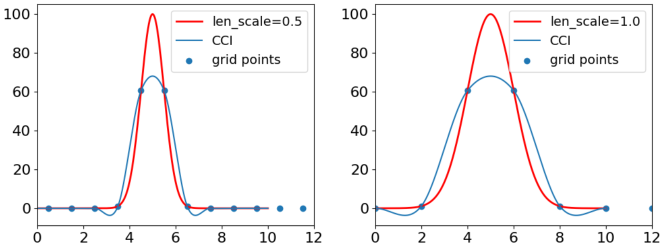
$$\rho = \frac{l}{d}, \tag{10}$$

and thus combines the kernel hyperparamters and the interpolation method, and thereby allowing us to adapt and converge at an optimal value for M_{opt}. See Section III-A.

We now aim to show that the introduction of ρ does not affect the accuracy of KISSGP (7). Let $\mathbf{a} \in \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M\}$, then we consider the task of interpolating the kernel value $k(\mathbf{x}_i, \mathbf{a})$ at $\mathbf{x}_i$, $1 \leq i \leq N$. Here, the distance from $\mathbf{x}_i$ to $\mathbf{a}$ can be expressed as function of d i.e., $\|\mathbf{x}_i - \mathbf{a}\|_2 = s_{xa} d = s_{xa}\frac{l}{\rho}$, where s_{xa} is the relative distance from $\mathbf{x}_i$ to $\mathbf{a}$, in terms of the distance between gridpoints d. Note that it is valid to represent a point $\mathbf{x}_i$ by its relative distance s_{xa} since the relationship is bijective when ρ and l are specified. Subsequently, the kernel value $k(\mathbf{x}_i, \mathbf{a})$ in (7) can be explicitly expressed as a function of ρ, i.e.,

$$k(\mathbf{x}_i, \mathbf{a}) = \sigma_s^2 \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{a}\|_2^2}{2l^2} \right) = \sigma_s^2 \exp\left(-\frac{s_{xa}^2}{2\rho^2} \right). \tag{11}$$

Along similar lines $k(\mathbf{u}_m, \mathbf{a}), 1 \leq m \leq M$, can also be expressed as a function of ρ, where the relative distance between grid points is 1 by definition. Thus the performance



(a) $l = 0.5$ (b) $l = 1.0$

Fig. 1: Comparison of interpolation result with $\rho = 1$. a) Using 16 grid points. b) Using 8 grid points.

of the linear operation in (7) should not be affected by the introduction of ρ.

Figure 1 gives an illustration for a 1-D case, which is obtained by fixing $\rho = 1$ to an approximate RBF kernel with $l = 0.5$ and $l = 1$. From the plots, we observe that both the true kernel value distribution and the interpolated value distribution are stretched along the x-axis, however the values remain unchanged. In summary, the criteria for accuracy can be uniquely controlled by the newly proposed parameter: *density* ρ, which directly yields the number of grid points M.

A. *MKISSGP*

We propose a framework that leverages grid point density to ensure accuracy at a reduced cost, and introduce a training algorithm where at every iteration, the grid points are updated according to ρ to adapt to the change in the length scale l. We consider here an RBF kernels and single dimension $D = 1$, but a similar density parameter can be learned for other kernels, and can be extended for larger dimensions. The steps of the training algorithm are summarized in Algorithm 1, and for each iteration we follow these tasks.

- **Determine grid points** Given the inputs $\mathbf{X}$, the length scale l, and density parameter ρ, we determine the distance between grid points as $d = \frac{l}{\rho}$. To guarantee that all training inputs are effectively covered, we add sufficient number of grid points at the beginning and the end with the same distance to ensure that the interpolation can be effectively achieved.
- **Calculate interpolation matrix (W)** Because the grid points now change with the optimization, the interpolation matrix $\mathbf{W}$ is recalculated in every iteration, unlike in standard KISSGP.
- **Calculate kernel ($\mathbf{K}_{\mathbf{U},\mathbf{U}}$)** Given the updated hyperparameters and selected grid points $\mathbf{U}$, the updated kernel matrix is calculated.
- **Update hyperparameters $\boldsymbol{\theta}$** Finally, we follow the steps of non-linear conjugate gradient (CG) approach to update the hyperparameters.

B. *Time complexity*

Compared with the original KISSGP, MKISSGP introduced the procedure of grid point determination and interpolation matrix calculation $\mathbf{W}$ at every iteration, which introduces additional time complexity terms. The determination of grid points takes $\mathcal{O}(M)$ time to evaluate the position of the M grid points in each dimension. Due to the fact that in the SKI framework, we assume a multiplicative kernel [20], and the CCI will require at least 4 grid points per dimension, $\mathcal{O}(MD) \leq \mathcal{O}(M^D)$. Secondly, for the calculation of the interpolation matrix $\mathbf{W}$, the required grid points and the corresponding weights for the training data need to be calculated, which both take $\mathcal{O}(N)$ time, but can be computed in parallel.

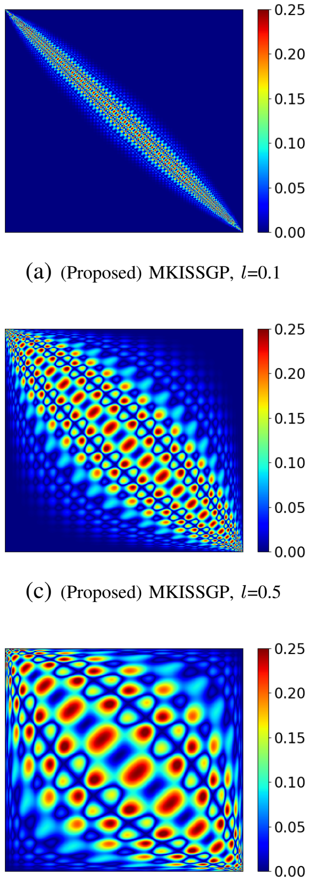
Algorithm 1 Training Process for MKISSGP

```
Input: Training set $\{\mathbf{X}, \mathbf{y}\}$, initial guess $\boldsymbol{\theta}_0$, density $\rho$
Determine grid points from $\rho$ and $l$
Calculate the initial $\mathbf{W}$, $\mathbf{K}_{\mathbf{U},\mathbf{U}}$
Define index $k \leftarrow 0$
Estimate $\boldsymbol{\Delta}_k \leftarrow \frac{-\partial \log p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta}_k)}{\partial \boldsymbol{\theta}_k}$
Define initial conjugate direction $\mathbf{s}_k \leftarrow \boldsymbol{\Delta}_k$
repeat
    Determine grid points from $\rho$ and $l$
    Update the interpolation matrix $\mathbf{W}$
    Update the grid kernel matrix $\mathbf{K}_{\mathbf{U},\mathbf{U}}$
    Estimate $\boldsymbol{\Delta}_{k+1} \leftarrow \frac{-\partial \log p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta}_{k+1})}{\partial \boldsymbol{\theta}_{k+1}}$ using $\mathbf{W}, \mathbf{K}_{\mathbf{U},\mathbf{U}}$
    Estimate $\beta_k \leftarrow \frac{\boldsymbol{\Delta}_{k+1}^T(\boldsymbol{\Delta}_{k+1}-\boldsymbol{\Delta}_k)}{\boldsymbol{\Delta}_k^T\boldsymbol{\Delta}_k}$
    Update conjugate direction: $\mathbf{s}_{k+1} \leftarrow \beta_k\mathbf{s}_k + \boldsymbol{\Delta}_{k+1}$
    Perform Wolfe line search: $\alpha_k \leftarrow \alpha$
    Update hyperparameter: $\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \alpha_k\mathbf{s}_k$
    Update index $k \leftarrow k+1$
until convergence
Output: Optimal hyperparameters $\boldsymbol{\theta}_{k-1}$
```

IV. Simulations

We perform 3 experiments to demonstrate the performance of the proposed MKISSGP in comparison with state-of-the-art.

Kernel reconstruction: We generate 3 RBF kernel matrices from 10000 random points drawn from a standard normal distribution while using RBF kernels with $l = 0.1$, $l = 0.5$, and $l = 1.0$, respectively. We approximate the kernels with a density $\rho = 2.7$ and KISSGP with $M = 42$, equivalent to MKISSGP in the $l = 0.5$ case. In Figure 2, we see the absolute reconstruction error, as compared to the true kernel matrix. In Figure 2a, 2c, and 2e, the error of reconstruction using MKISSGP is shown, where the values of the errors remain relatively consistent with the length scale variation. In Figure 2b, 2d, and 2f, we see the error of reconstruction using KISSGP, where the error significantly scales with smaller values of l and constant M. For $l = 0.5$, despite having the same number of grid points, the reconstruction error differs between MKISSGP and KISSGP, due to differences in grid point position selection. This experiment confirms that by choosing higher values of ρ, one can improve the accuracy of kernel matrix reconstruction without overestimating the required number of grid points M.



(a) (Proposed) MKISSGP, l=0.1

(b) KISSGP, l=0.1

(c) (Proposed) MKISSGP, l=0.5

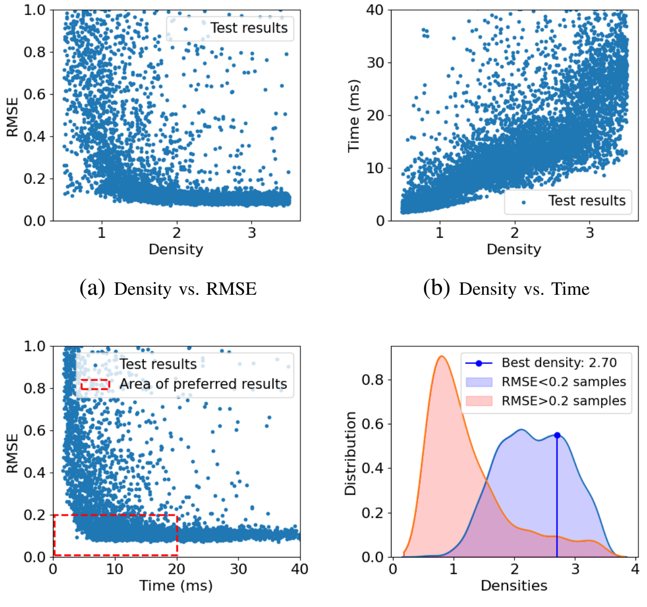
(d) KISSGP, l=0.5

(e) (Proposed) MKISSGP, l=1.0

(f) KISSGP, l=1.0

Fig. 2: Reconstruction error between true and approximated kernel matrix $|\mathbf{K} - \tilde{\mathbf{K}}|$, where $|\cdot|$ denotes the element-wise absolute value. The color bar is limited to 0.25, as values may be larger.

Recommended Density (ρ)*:* In the next experiment, our objective is to determine an optimal density value for MKISSGP using the RBF. We perform 8,000 Monte Carlo trials, where in every trial, we generate 1,000 noisy training points (with $\sigma_n^2 = 0.25$) from an arbitrarily sampled $D = 1$ function governed by a Gaussian process (GP) using an RBF kernel. The signal power is drawn from a uniform distribution ranging between 1.0 and 10.0, while the length scale is selected from a uniform distribution in the logarithmic domain spanning from 0.1 to 20.0. For each trial, a MKISSGP model is constructed with densities drawn from a uniform distribution between 0.5 and 3.5. The efficiency of the algorithm is gauged by recording the average time spent on one negative marginal log-likelihood (NMLL) and derivative of NMLL w.r.t. hyperparameters calculation, along with the final RMSE achieved.



(a) Density vs. RMSE

(b) Density vs. Time

(c) Time vs. RMSE

(d) Distribution of samples

Fig. 3: Recommended density test results: (a) root mean square error (RMSE) decreases with increasing ρ, (b) Time generally rises with density. (c) RMSE decreases with time; preferred results are indicated (time $<$ 20ms, RMSE $<$ 0.2). (d) KDE plots. Blue: preferred samples (RMSE $<$ 0.2). Red: samples with RMSE $>$ 0.2. Samples are weighted considering error and time.

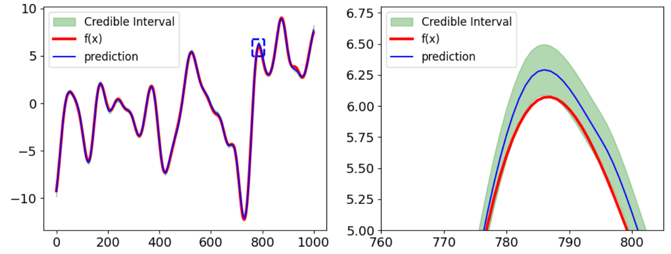


Fig. 4: Function $f(x)$ with posterior distribution obtained using MKISSGP with $\rho = 2.7$.

Figure 3a and 3b show the error and time across density values ρ, respectively. The covariance of the error distribution decreases, and the time spent increases with growing density. The scattering of time values can be attributed to the stochastic nature of computational power and variations in hyperparameter initialization. The relationship between time per iteration and the RMSE is shown in Figure 3c. Achieving better results is more likely when there is a higher time used. The region within the red (dotted) box is considered where the optimal test results are expected to be found, balancing time and RMSE. Beyond the threshold of 20 ms, minimal improvement in results is observed and an upper bound of 0.2 is imposed on the RMSE. Figure 3d shows the distribution of samples with RMSE values less and greater than 0.2, where we have chosen $\boldsymbol{\rho} = 2.70$, as it maximizes the difference between the kernel density estimate (KDE) values. Individual requirements and unique characteristics of the data may require adjustments to the chosen density.

Computational complexity: We now generate a function

Method	$\bar{\rho}$	RMSE	time (ms)
GPR	-	0.11	8601.63
KISSGP-50	1.41	0.25	1201.94
KISSGP-100	2.91	0.13	4793.28
KISSGP-200	5.91	0.11	6727.39
MKISSGP-2.2	2.20	0.15	947.71
MKISSGP-2.7	**2.70**	**0.11**	**1155.03**
MKISSGP-3.2	3.20	0.11	1523.23

TABLE I: RMSE and time of GPR, KISSGP and MKISSGP, where $\bar{\rho}$ indicates equivalent density for KISSGP.

$f(x)$, using an RBF kernel with parameters $\sigma_s^2 = 25$, $l = 30$, and the underlying noise variance of $\sigma_n^2 = 0.25$, using $N = 1000$ samples as shown in Figure 4. We compare the reconstruction error of GPR, KISSGP and MKISSGP, for various equivalent density parameters, and all experiments were run on a 2.30GHz Intel Core i7-11800H CPU. Table I shows the RMSE and training time for the three methods reconstructing function $f(x)$ over 100 Monte Carlo trials. We observe that to achieve our benchmark accuracy of the GPR, KISSGP requires $M = 200$ gridpoints at an equivalent density of 5.91, while MKISSGP only needs a density of 2.7 to achieve a similar accuracy, while requiring almost 6 times less training time.

The asymptotic time complexity of MKISSGP is found to be reduced to $\mathcal{O}(N + M_{opt}^2)$, where M_{opt} represents the optimal number of grid points to reach a specific level of accuracy given the length scale l. During nonlinear conjugate gradients (CG), the changes in the length scale become progressively smaller to the extent that the number of grid points remains unchanged. To achieve the desired accuracy, our proposed MKISSGP algorithm converged at $M_{\text{opt}} = 81$ with $\rho = 2.7$, while using the state-of-the-art KISSGP, the corresponding accuracy was only reached at $M = 200$, while traditional GPR uses $N = 1000$ points. As mentioned earlier, we could further reduce this complexity to $\mathcal{O}(N + M_{opt} \log M_{opt})$ if we exploit the underlying Toeplitz structure [15].

V. Conclusion

In this work, we presented MKISSGP, a malleable extension of KISSGP, where the grid points can be computed dynamically. We show that MKISSGP effectively minimizes the number of grid points required to achieve desired accuracy, as compared to the state-of-the-art methods, and thus reduces computational complexity. The benefits of our strategy are corroborated with extensive simulations. In this work, we limited our discussion to RBF kernels and CCI, and in our follow up work, we aim to explore other kernels and interpolation methods, and recommend suitable density parameters in these scenarios.

Acknowledgement: This work is partially funded by EU HORIZON-CHIPS-JU-2023-2-RIA, under grant agreement No 101139996, the ShapeFuture project.

References

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

[2] F. Pérez-Cruz, S. Van Vaerenbergh, J. J. Murillo-Fuentes, M. Lázaro-Gredilla, and I. Santamaria, "Gaussian processes for nonlinear signal processing: An overview of recent advances," *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 40–50, 2013.

[3] R. Krems, "Bayesian machine learning for quantum molecular dynamics," *Physical Chemistry Chemical Physics*, vol. 21, no. 25, pp. 13 392–13 410, 2019.

[4] V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti, and G. Csányi, "Gaussian process regression for materials and molecules," *Chemical Reviews*, vol. 121, no. 16, pp. 10 073–10 141, 2021.

[5] Y. Pan, X. Yan, E. A. Theodorou, and B. Boots, "Prediction under uncertainty in sparse spectrum gaussian processes with applications to filtering and control," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2760–2768.

[6] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using gaussian process regression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2019.

[7] M. Jadaliha, Y. Xu, J. Choi, N. S. Johnson, and W. Li, "Gaussian process regression for sensor networks under localization uncertainty," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 223–237, 2012.

[8] A. G. Wilson, D. A. Knowles, and Z. Ghahramani, "Gaussian process regression networks," *arXiv preprint arXiv:1110.4411*, 2011.

[9] A. Damianou and N. D. Lawrence, "Deep gaussian processes," in *Artificial intelligence and statistics*. PMLR, 2013, pp. 207–215.

[10] M. Deisenroth and J. W. Ng, "Distributed gaussian processes," in *International conference on machine learning*. PMLR, 2015, pp. 1481–1490.

[11] A. Venkitaraman, S. Chatterjee, and P. Handel, "Gaussian processes over graphs," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5640–5644.

[12] P. Zhai and R. T. Rajan, "Distributed gaussian process hyperparameter optimization for multi-agent systems," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[13] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When gaussian process meets big data: A review of scalable GPs," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 11, pp. 4405–4423, 2020.

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

[15] M. Yadav, D. Sheldon, and C. Musco, "Faster kernel interpolation for gaussian processes," *International Conference on Artificial Intelligence and Statistics*, pp. 2971–2979, 2021.

[16] M. Yadav, D. R. Sheldon, and C. Musco, "Kernel interpolation with sparse grids," *Advances in Neural Information Processing Systems*, vol. 35, pp. 22 883–22 894, 2022.

[17] C. Williams and M. Seeger, "Using the nyström method to speed up kernel machines," *Advances in neural information processing systems*, vol. 13, 2000.

[18] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," *Advances in neural information processing systems*, vol. 18, 2005.

[19] M. Lázaro-Gredilla, J. Quinonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal, "Sparse spectrum gaussian process regression," *The Journal of Machine Learning Research*, vol. 11, pp. 1865–1881, 2010.

[20] A. Wilson and H. Nickisch, "Kernel interpolation for scalable structured gaussian processes (KISS-GP)," *International conference on machine learning*, pp. 1775–1784, 2015.

[21] A. G. Wilson, C. Dann, and H. Nickisch, "Thoughts on massively scalable gaussian processes," *arXiv preprint arXiv:1511.01870*, 2015.

[22] P. Izmailov, A. Novikov, and D. Kropotov, "Scalable gaussian processes with billions of inducing inputs via tensor train decomposition," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2018, pp. 726–735.

[23] A. Smola and P. Bartlett, "Sparse greedy gaussian process regression," *Advances in neural information processing systems*, vol. 13, 2000.

[24] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE transactions on acoustics, speech, and signal processing*, vol. 29, no. 6, pp. 1153–1160, 1981.