Bayesian Fourier Features for Reduced Rank Gaussian Processes

Cristian A. Galvis-Florez
Aalto University
cristian.galvis@aalto.fi

George Whittle
Machine Learning Research Group,
University of Oxford

Michael A. Osborne
Machine Learning Research Group,
University of Oxford

Simo Särkkä
ELLIS Institute Finland,
Aalto University

Abstract

Gaussian processes are probabilistic models used in machine learning and the physical sciences, although they are limited by cubic complexity in the number of training observations. To mitigate this problem, various low-rank kernel approximation methods, including Fourier feature methods, Hilbert space methods, and inducing point methods, have been developed.

In this paper, we propose a novel Fourier feature approach leveraging Bayesian quadrature methods to construct reduced-rank approximations of the Gaussian process kernel. The new Bayesian Fourier feature framework also unifies many previously proposed low-rank methods, as they can be seen as instances of Bayesian quadrature-based approximations of Gaussian process kernels. Due to its probabilistic nature, the unified framework also enables the quantification of uncertainty in the approximation. Furthermore, the framework allows for the design of entirely new low-rank kernel approximations.

We compare the performance of the proposed methods with other approaches across different kernel length scales. Our experimental results demonstrate that it outperforms other popular low-rank kernel approximation methods across a wide range of length scales.

Proceedings of the 29[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

1 INTRODUCTION

Gaussian processes (GPs) are a flexible Bayesian framework for modeling multidimensional, nonlinear functions, with applications in machine learning (Rasmussen et al., 2006), physical sciences (Deringer et al., 2021), spatial statistics (Hamelijnck et al., 2021), and numerical integration through Bayesian quadrature (O'Hagan, 1991; Minka, 2000; Hennig et al., 2022). They are characterized by their ability to provide principled uncertainty quantification for predictions. The kernel function captures the dependencies between function values at different input points, playing a crucial role in the modeling process. However, GPs face computational challenges due to the cubic computational complexity $O(N^3)$ in the number of training observations N, which results from the need to invert the $N \times N$ kernel matrix. To mitigate this, different low-rank kernel approaches have been proposed to approximate the kernel matrix, leading to a complexity reduction for GP applications.

Low-rank Gaussian process methods construct approximations to the kernel matrix that reduce the computational cost of inference. These approaches include inducing point methods (Quinonero-Candela and Rasmussen, 2005; Titsias, 2009; Snelson and Ghahramani, 2005; Chalupka et al., 2013), Fourier feature methods that can be constructed using either random sampling (Rahimi and Recht, 2007; Lázaro-Gredilla et al., 2010) or quadrature rules (Dao et al., 2017; Mutny and Krause, 2018; Shustin and Avron, 2022; Li et al., 2024), and eigenfunction or Hilbert space decompositions (Williams and Seeger, 2000; Solin and Särkkä, 2020; Greengard and O'Neil, 2022).

In this paper, we specifically consider Fourier feature-based methods. They can be derived from Bochner's theorem, which states that any stationary kernel can

be expressed as the Fourier transform of its spectral density (Akhiezer and Glazman, 1993). In Fourier feature methods, the integral is approximated either by Monte Carlo sampling, as in Random Fourier Features (RFF) (Rahimi and Recht, 2007), or by deterministic quadrature schemes, such as Gauss–Hermite (GH) (Mutny and Krause, 2018), Gauss–Legendre (GL) (Shustin and Avron, 2022), or trigonometric quadratures (Li et al., 2024). Hilbert space methods (Solin and Särkkä, 2020) instead use eigenfunction expansions to approximate the kernel, which in some instances leads to the Fourier basis functions. These low-rank representations replace the full $N \times N$ kernel matrix with one of rank M, leading to a reduced computational complexity of $O(NM^2)$, which is substantially cheaper than $O(N^3)$ when $M \ll N$.

GPs can be used for numerical integration to build Bayesian quadrature rules (O'Hagan, 1991; Minka, 2000; Hennig et al., 2022). Bayesian quadratures stand out from classical quadrature rules because they provide principled uncertainty quantification for the integral and can also consider noisy measurements of the function that is integrated. Classical quadrature methods can be seen as instances of Bayesian quadratures (Särkkä et al., 2014, 2016; Karvonen and Särkkä, 2017; Hennig et al., 2022). This motivates the main goal of this paper, which is to propose a Bayesian quadrature approach to solve Bochner's integral and construct uncertainty-aware low-rank approximations for stationary kernels.

The contribution of the paper is to introduce a new framework, where we replace the deterministic Fourier features with their probabilistic counterparts that we call *Bayesian Fourier features* (BFF). The framework also unifies many previously proposed low-rank methods, which are instances of Bayesian quadrature-based approximations of Gaussian process kernels, such as the Gauss-Hermite and Gauss-Legendre-based methods. Because of its probabilistic nature, the unified framework also facilitates the quantification of uncertainty in the approximation. Additionally, it enables the development of entirely new low-rank kernel approximations by changing the kernel or the nodes of the Bayesian quadrature.

We compare the performance of the proposed method with other approaches across different kernel length scales, addressing one of the main challenges for low-rank kernel approximations that work in a limited range of the length scale parameter. Our experimental results demonstrate that it outperforms other popular low-rank kernel approximation methods across a wide length scale range.

The structure of the paper is as follows: In Section 2, we review the formulation of Gaussian process regression (GPR) and quadrature (GPQ) applications. Also, we review their low-rank approximations and applications in GPR. We provide the BFF method for kernel approximation in Section 3 and give a detailed implementation in Section 4. The proposed low-rank method is implemented for GPR in Section 5. We then conclude our findings in Section 6.

2 PRELIMINARIES

In this section, we review the formulation of GPs and their implementation for Gaussian process regression (GPR) and Gaussian process quadrature (GPQ). Afterwards, we review the notation of low-rank GPs and how they provide advantages in terms of complexity when applied to GPR.

Gaussian processes, denoted here by $f \sim \mathcal{GP}\left(\mu(\mathbf{x}), k\left(\mathbf{x}, \mathbf{x}'\right)\right)$, can be used to model and predict the behavior of an $\mathbb{R}^d \mapsto \mathbb{R}$ function $f(\mathbf{x})$ by modeling its evaluations as jointly Gaussian. GP regression amounts to conditioning the prior GP with the given mean and covariance to noisy observations from the model $y_i = f\left(\mathbf{x}_i\right) + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ at the points $\mathbf{x}_i$, which forms a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i) \ : \ i = 1, \ldots, N\}$. Gaussian process regression intends to provide an approximation of the function $f(\mathbf{x})$ on an unseen input point $\mathbf{x}_*$ given the dataset $\mathcal{D}$. The prediction is characterized by the Gaussian posterior distribution $p\left(f_* \mid \mathbf{x}_*\right) = \mathcal{N}\left(f_* \mid \mathrm{E}[f_*], \mathrm{V}[f_*]\right)$ with mean and variance given by (Rasmussen et al., 2006)

$$\begin{aligned} \mathrm{E}[f_*] &= \mathbf{k}_*^\top \left(\mathbf{K} + \sigma_n^2 I\right)^{-1} \mathbf{y}, \\ \mathrm{V}[f_*] &= k\left(\mathbf{x}_*, \mathbf{x}_*\right) - \mathbf{k}_*^\top \left(\mathbf{K} + \sigma_n^2 I\right)^{-1} \mathbf{k}_*, \end{aligned} \tag{1}$$

where $\mathbf{K}$ is the $N \times N$ covariance matrix with elements $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ representing covariances between training inputs, $\mathbf{k}_*$ is the covariance vector with entries $k(\mathbf{x}_*, \mathbf{x}_i)$ for the new input, and $\mathbf{y} = (y_1, \ldots, y_N)$ is a vector formed of the measurements.

Gaussian processes can also be used for numerical integration via Bayesian quadrature rules. In this case, the method is known as Gaussian process quadrature (GPQ). GPQs, along with the classical quadrature rules, attempt numerical computation of the integral

$$\mathcal{I} = \int_\Omega f(\mathbf{x}) \mu(\mathbf{x}) d\mathbf{x} \tag{2}$$

for a given function $f(\mathbf{x})$ and weight function $\mu(\mathbf{x})$. In this paper, we will later connect this integral to Bochner's integral. In the GPQ context, the function f is modeled using a Gaussian process defined by a kernel function $k_Q(\mathbf{x}, \mathbf{x}')$. As a result, measurements $\mathbf{y}$ at nodes $\mathcal{X}$ can be used to form an estimator for the

integral $\mathcal{I}$ with mean and variance given by (O'Hagan, 1991; Minka, 2000)

$$\begin{aligned} Q_{\mathrm{BQ}} &= k_{Q_\mu}(\mathcal{X})^\top \mathbf{K}_Q^{-1}\mathbf{y}, \\ V_{\mathrm{BQ}} &= \mu(k_{Q_\mu}) - k_{Q_\mu}(\mathcal{X})^\top \mathbf{K}_Q^{-1} k_{Q_\mu}(\mathcal{X}). \end{aligned} \tag{3}$$

Here, the ith entry of the vector $k_{Q_\mu}(\mathcal{X})$ is defined as $k_{Q_\mu}(\mathbf{x}_i) = \int_\Omega k_Q(\mathbf{x}_i, \mathbf{x}')\mu(\mathbf{x}')d\mathbf{x}'$, and the scalar $\mu(k_{Q_\mu})$ is given by $\mu(k_{Q_\mu}) = \int_\Omega \int_\Omega k_Q(\mathbf{x}, \mathbf{x}')d\mu(\mathbf{x})d\mu(\mathbf{x}')$. The quadrature Q_{BQ} estimates the integral $\mathcal{I}$ using the often closed-form integral of the kernel function, avoiding the need to integrate f directly.

Evaluating the posterior mean and variance of either GPR or GPQ requires a computational complexity $O(N^3)$, which is inefficient for large data sets. To solve this problem, several approximation methods have been proposed that reduce the computational complexity of the regression.

Low-rank Gaussian processes: In Gaussian process regression and quadratures, it is common to use stationary covariance functions that satisfy the stationary property, which means that $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$. Taking $\mathbf{r} = \mathbf{x} - \mathbf{x}'$ the kernel is just a function of the difference $\mathbf{r}$, and the Bochner's theorem allows us to write $k(\mathbf{r}) = \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d} S(\boldsymbol{\omega})e^{i\boldsymbol{\omega}^\top \mathbf{r}}d\boldsymbol{\omega}$, where $S(\boldsymbol{\omega})$ is the spectral density of the kernel (Akhiezer and Glazman, 1993). With real-valued data and a kernel, the spectral density is symmetric with respect to the origin. The antisymmetric imaginary part of the integral vanishes, leading to

$$k(\mathbf{x}, \mathbf{x}') = \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d} S(\boldsymbol{\omega})\cos(\boldsymbol{\omega}^\top \mathbf{r})d\boldsymbol{\omega}. \tag{4}$$

If we see this as an integral in the form Eq. (2), a quadrature rule provides an approximation of the integral in Eq. (4) in the form

$$k(\mathbf{x}, \mathbf{x}') \approx \sum_{j=1}^{M} w_j \cos(\boldsymbol{\xi}_j^\top(\mathbf{x} - \mathbf{x}')), \tag{5}$$

where $\boldsymbol{\xi}_j$ represent the nodes and w_j the weights of the quadrature rule. There are several different ways to split between the function f and the weight function μ in Eq. (2). One way would be to set $\mu(\omega) = S(\omega)/(2\pi)^d$ and $f(\omega) = \cos(\boldsymbol{\omega}^\top \mathbf{r})$, which may be convenient but generally leads to intractable integrals in forming the GPQ. Therefore, in this paper we focus on the choices $\mu(\omega) = 1$ and $f(\omega) = S(\boldsymbol{\omega})\cos(\boldsymbol{\omega}^\top \mathbf{r})/(2\pi)^d$ which due to the complete freedom of choosing the GPQ covariance function provide practically the same flexibility as the non-unity weight function can always be encoded into the covariance kernel of the GPQ.

Using trigonometric identities and defining the vector

$$\begin{aligned} \boldsymbol{\phi}(\mathbf{x}) = (&\cos(\boldsymbol{\xi}_1^\top \mathbf{x}), \cdots, \cos(\boldsymbol{\xi}_M^\top \mathbf{x}), \\ &\sin(\boldsymbol{\xi}_1^\top \mathbf{x}), \cdots, \sin(\boldsymbol{\xi}_M^\top \mathbf{x}))^\top, \end{aligned} \tag{6}$$

and the matrix $\Lambda = \mathrm{diag}(w_1, \ldots, w_M, w_1, \ldots, w_M)$, we can write Eq. (4) in the form

$$k(\mathbf{x}, \mathbf{x}') \approx \boldsymbol{\phi}^\top(\mathbf{x})\Lambda\boldsymbol{\phi}(\mathbf{x}'). \tag{7}$$

The equation above expresses the kernel in a form that is characteristic to common low-rank Gaussian process approximations (Lázaro-Gredilla et al., 2010; Solin and Särkkä, 2020; Shustin and Avron, 2022; Li et al., 2024), allowing a computational complexity reduction in Gaussian process applications. The form of the approximation also ensures that it is positive definite. More generally, the definition of the column vector $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^{2M}$ components might differ according to the approximation used.

Once the kernel is written in the low-rank representation (7), it can be used for low-rank Gaussian process regression. Using the matrix inversion lemma (Woodbury, 1950), the mean and variance that define the posterior distribution in Eq. (1) can be written as

$$\begin{aligned} \mathrm{E}[f_*] &\approx \boldsymbol{\phi}_*^\top \left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \sigma_n^2\Lambda^{-1}\right)^{-1}\boldsymbol{\Phi}^\top\mathbf{y}, \\ \mathrm{V}[f_*] &\approx \sigma_n^2\boldsymbol{\phi}_*^\top \left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \sigma_n^2\Lambda^{-1}\right)^{-1}\boldsymbol{\phi}_*. \end{aligned} \tag{8}$$

Here, $\boldsymbol{\phi}_* = \boldsymbol{\phi}(\mathbf{x}_*)$ represents the feature vector associated with the new input data point $\mathbf{x}_*$. The ith row of the matrix $\boldsymbol{\Phi} \in \mathbb{R}^{N\times 2M}$ contains the components of the vector $\boldsymbol{\phi}(\mathbf{x}_i)$ corresponding to the input data point $\mathbf{x}_i$. The inverse in Eq. (8) is now over a matrix of size $2M \times 2M$, however, symmetries in the nodes $\boldsymbol{\xi}_j$ can reduce it to a $M \times M$ matrix (Karvonen et al., 2019). Since we choose $M < N$, we have an advantage with respect to ordinary GPs methods.

Different approximation methods differ in how the components vector $\boldsymbol{\phi}(\mathbf{x})$ and the matrix Λ are defined. RFF considers a Monte Carlo approximation of Eq. (4) by taking samples of the spectral density (Lázaro-Gredilla et al., 2010). Hilbert space (HS) approximation methods approximate the kernel by considering a Hilbert space $\mathcal{H}$ defined over a domain Ω in $\mathbb{R}^N$ with a set of orthonormal basis functions $\{\varphi_i(\mathbf{x})\}$ and corresponding eigenvalues $\{\lambda_i\}$ (Solin and Särkkä, 2020). Gaussian quadrature methods can also be used to approximate the kernel (Dao et al., 2017). The squared exponential kernel naturally leads to Gauss–Hermite Fourier features (GHFF) (Mutny and Krause, 2018).

If we truncate the integral in Eq. (4) to a finite domain Ω, it is possible to define a kernel approximation based

on Gauss–Legendre Fourier features (GLFF) (Shustin and Avron, 2022). Also, under the truncation idea, it is possible to approximate the kernel using a trigonometric quadrature rather than a Gaussian one, as shown by Li et al. (2024) using trigonometric quadrature Fourier features (TQFF).

The existing Fourier feature-based methods for kernel approximation employ classical quadrature techniques to estimate the integral in Eq. (4). Therefore, in the following, we refer to them as classical Fourier features.

3 BAYESIAN FOURIER FEATURES FOR KERNEL APPROXIMATION

In this section, we introduce a Bayesian quadrature approach for kernel approximation, resulting in a novel methodology that we call Bayesian Fourier features (BFFs) for kernel approximation.

Classical quadrature methods can be seen as special instances of Bayesian quadratures (Särkkä et al., 2014, 2016; Karvonen and Särkkä, 2017; Hennig et al., 2022). This motivation leads to the main contribution of this paper, which proposes a Bayesian quadrature method to approximate the kernel using Bochner's theorem (4). As a result, we get a class of BFFs for reduced-rank kernel approximation.

Let us first consider the one-dimensional case and a stationary kernel $k(r)$ with $r = x - x'$. This implies that the spectral density satisfies $S(\omega) = S(-\omega)$ and the Bochner's integral takes the form

$$k(r) = \frac{1}{2\pi} \int_{\mathbb{R}} S(\omega) \cos(\omega r)\, d\omega. \tag{9}$$

We can now implement a Bayesian quadrature for this integral using a set of nodes $\Xi = \{\xi_i\}_{i=1}^M$ and corresponding evaluations $\boldsymbol{\eta} = \{f(\xi_i)\}_{i=1}^M$ of the integrand $f(\omega) = S(\omega)\cos(\omega r)$. The quadrature is constructed using the covariance function $k_Q(\omega, \omega')$, different from the kernel $k(r)$ that we want to approximate. Note that we could also choose $\mu(\omega) = S(\omega)$ and $f(\omega) = \cos(\omega r)$, but that would limit the generality of the method and is mathematically equivalent to a specific choice of k_Q.

Within the Bayesian quadrature framework, $S(\omega)\cos(\omega r)$ is modeled as a Gaussian process with covariance function $k_Q(\omega, \omega')$. Then, a Bayesian quadrature for Eq. (9) delivers $k(r) \sim \mathcal{N}\left(k(r) \mid \hat{k}(r), \Sigma\right)$, where $\hat{k}(r)$ and Σ denote the posterior mean and variance provided by Eqs. (3) leading to

$$\begin{aligned} \hat{k}(r) &= k_{Q_\mu}^\top(\boldsymbol{\Xi}) K_Q^{-1} \boldsymbol{\eta}, \\ \Sigma &= \mu(k_{Q_\mu}) - k_{Q_\mu}(\boldsymbol{\Xi})^\top \mathbf{K}_Q^{-1} k_{Q_\mu}(\boldsymbol{\Xi}). \end{aligned} \tag{10}$$

The kernel above corresponds to the Bayesian Fourier feature counterpart for the classical Fourier features approximation. The connection can be seen by writing the BFF mean as

$$\begin{aligned} \hat{k}(r) &= \sum_{i=1}^M \left[k_{Q_\mu}^\top(\boldsymbol{\Xi}) K_Q^{-1} \right]_i f(\xi_i) \\ &= \sum_{i=1}^M w_i^Q \cos(\xi_i r), \end{aligned}$$

where the weights are given by $w_i^Q = \left[k_{Q_\mu}^\top(\boldsymbol{\Xi}) \mathbf{K}_Q^{-1} \right]_i S(\xi_i)$. In contrast to the classical approximation, the quadrature formulation yields a posterior variance Σ associated with this low-rank representation, which delivers an uncertainty measure for reduced-rank kernel approximations.

An important insight arises by introducing the $2M$-dimensional feature map in Eq. (6) together with the matrix $\Lambda = \mathrm{diag}(w_1^Q, \cdots, w_M^Q, w_1^Q, \cdots, w_M^Q)$. With this definition, the posterior mean of the kernel can be written as

$$\hat{k}(x, x') = \boldsymbol{\phi}^\top(x) \Lambda \boldsymbol{\phi}(x'), \tag{11}$$

which recovers the low-rank kernel approximation in (7). The computational complexity of computing the weights in our algorithm is $O(M^3)$, which is the same as in, for example, inducing point methods or Hilbert space methods (e.g., Solin and Särkkä, 2020). However, the weights are calculated once and then stored for later use. Furthermore, our algorithm provides an uncertainty estimate for evaluations of the kernel approximation, acting as a measure of approximation quality. Once the weights $\{w_i^Q\}_{i=1}^M$ are computed, the resulting BFF reduced-rank kernel can be deployed in Gaussian process models, yielding an $O(M^3)$ complexity reduction from $O(N^3)$ for GPs applications. Algorithm 1 summarizes the implementation in the one-dimensional case.

Algorithm 1 Kernel approximation with BFF

Input: Spectral density $S(\omega)$, BQ kernel $k_Q(\omega, \omega')$, nodes $\Xi = \{\xi_i\}_{i=1}^M$, domain Ω.
Output: Kernel approximation $\hat{k}(x, x')$ with variance $\Sigma\mu(k_\mu) - k_\mu^\top \mathbf{K}_Q^{-1} k_\mu$.

1: Build $[\mathbf{K}_Q]_{ij} \leftarrow k_Q(\xi_i, \xi_j)$.
2: Compute $[k_\mu]_i$ and $\mu(k_\mu)$.
3: Solve $\mathbf{b} \leftarrow k_\mu^\top \mathbf{K}_Q^{-1}$.
4: Set $w_i^Q \leftarrow b_i\, S(\xi_i)$, for $i = 1, \ldots, M$.
5: Compute $\Lambda = \mathrm{diag}(w_1, \ldots, w_M, w_1, \ldots, w_M)$.
6: Compute the vectors $\phi(x)$. ▷ Use Eq. (6)
7: Mean: $\hat{k}(x, x') \leftarrow \boldsymbol{\phi}^\top(x) \Lambda \boldsymbol{\phi}(x')$.
8: Variance: $\Sigma \leftarrow \mu(k_\mu) - k_\mu^\top \mathbf{K}_Q^{-1} k_\mu$.

The implementation of Bayesian quadrature for the integral in (4) depends on the choice of covariance function k_Q and the selection of quadrature nodes Ξ. In practice, one may also truncate the integration domain to avoid integrating over an infinite range.

We can also consider the multi-dimensional case where the data points $\mathbf{x} \in \mathbb{R}^d$ have components $\{x_i\}_{i=1}^d$. If the kernel can be written as the product of one-dimensional stationary kernels $k(\mathbf{x}, \mathbf{x}') = k_1(x_1, x_1') \times k_2(x_2, x_2') \times \cdots \times k_d(x_d, x_d') = \prod_{i=1}^d k_i(x_i - x_i')$, we can implement the BFF method in each dimension. This multidimensional extension suffers from the well-known curse of dimensionality. This can be overcome by exploiting symmetry properties of the nodes or by sparse-grid integration (Karvonen and Särkkä, 2018; Gerstner and Griebel, 1998).

4 IMPLEMENTATION FOR SQUARED EXPONENTIAL KERNEL

In this section, we show how to implement the method for a one-dimensional squared exponential kernel.

We consider the kernel $k(x, x') = \alpha \exp(-r^2/2\sigma^2)$ with hyperparameters α and σ. The corresponding spectral density is $S(\omega) = \alpha\sqrt{2\pi\sigma^2}\exp(-(\sigma\omega)^2/2)$. Let us define the normalized spectral density function $p(\omega) = (2\pi)^{-1/2}\exp(-\omega^2/2)$ independent of the hyperparameters. The spectral density can then be rewritten as $S(\omega) = 2\pi\alpha\sigma p(\sigma\omega)$ and we can write the integral in (9) as

$$k(x, x') = \alpha \int_{-\infty}^{\infty} p(\omega) \cos\left(\omega \left[\frac{x - x'}{\sigma}\right]\right) d\omega \qquad (12)$$

The method for constructing the BFF may vary depending on the choice of nodes, integration limits, and quadrature kernel. Here, we propose a squared exponential kernel for the quadrature $k_Q = \alpha_Q \exp(-(\omega - \omega')^2/2\sigma_Q^2)$, and we consider four approaches for this example in which the domain of integration and nodes change.

In the first approach, we choose the integration interval as $(-\infty, \infty)$ and use the roots of the Hermite polynomial of degree $2M$ as the nodes of the quadrature. Similarly to a Gauss–Hermite (GH) quadrature rule, we refer to this as GH–BFF. For the next approaches, we consider an integration domain $[-L, L]$; in the second approach, we divide the interval into $2M$ equidistant points, referring to this as the ED–BFF, meanwhile in our third approach, we implement a Gauss–Legendre (GL) quadrature with the roots of the Legendre polynomial of degree $2M$ as nodes of the integral approximation, and we refer to this as the GL–BFF approach. In the last approach, we consider a truncation over a γ parameter, as proposed in Li et al. (2024) for trigonometric Fourier features. Under this approach, the kernel is approximated as

$$k(x, x') \approx \alpha \int_{-\pi}^{\pi} p_\gamma(\gamma\omega) \cos\left(\omega\gamma \left[\frac{x - x'}{\sigma}\right]\right) d\omega, \quad (13)$$

with $p_\gamma = \gamma p(\gamma\omega)$, which we refer to as the γ–BFF method. For this approach, the set of nodes corresponds to the roots of the $2M$ Chebyshev polynomial for trigonometric quadratures.

For example, for the γ-BFF approach, we start by considering the function $f(\omega) = p_\gamma(\gamma\omega)\cos\left(\omega\gamma\left[\frac{x-x'}{\sigma}\right]\right)$ as a Gaussian process characterized by zero mean and a covariance function $k_Q(\omega, \omega')$. This allows us to define a Bayesian quadrature with nodes $\Xi = \{\xi_i\}_{i=1}^M$ given by $\xi_k = \cos(\pi(2k+1)/2M)$ and with its respective evaluations $\boldsymbol{\eta} = \{f(\xi_i)\}_{i=1}^M$.

Given this, the posterior distribution of the kernel $k(r)$ is now characterized by a Gaussian distribution $k(r) \sim \mathcal{N}\left(k(r) \mid \hat{k}(r), \Sigma\right)$ following the BFF procedure in the previous section. The mean and variance of this distribution are given by Eq. (3). The posterior mean of the BFF can be written as

$$\hat{k}(r) = \sum_{i=1}^{m} w_i^Q \cos(\xi_i r),$$

where the weights are given by $w_i^Q = \left[{k_Q}_\mu^\top(\Xi)\mathbf{K}_Q^{-1}\right]_i g(\Theta) p_\gamma(\gamma\xi_i)$, being these the weights of the γ-BFF approach. At this point, we can write the kernel in the low rank form in Eq. (7). We compare our method using the γ-BFF approach with popular low-rank methods in the literature, such as RFF (Lázaro-Gredilla et al., 2010), GHFF (Mutny and Krause, 2018), GLFF (Shustin and Avron, 2022), and the HS low-rank approximation (Solin and Särkkä, 2020).

We implement the method to approximate a squared exponential kernel with hyperparameters $\alpha = 1.5$ and $\sigma = 0.5$ over the interval $r \in [0, 8]$. Our γ-BFF method has kernel hyperparameters $\alpha_Q = 2S(0)$, $\sigma_Q = 0.3$ and $\gamma = 1.8$. The results in Figure 1 show the performance of our method for $M \in \{6, 14, 30\}$ features, including the root mean squared error (RMSE) for every method with respect to the real kernel. We observe that the γ-BFF approach outperforms the Fourier feature approaches presented here for fewer features, with an error similar to that of the HS method for this setup. An important result is that our BFF method gives us an uncertainty estimator for the approximation, which can be seen in Figure 1 since the BFF method is the

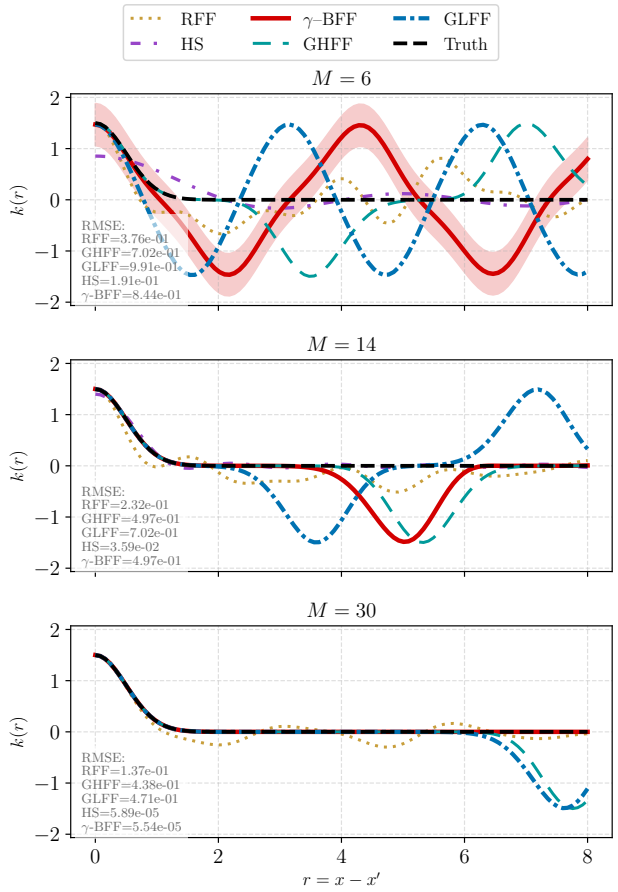


Figure 1: Approximation of the squared exponential kernel (dashed black) using RFF (dotted gold), GHFF (dashed teal), GLFF(dash-dotted blue), low-rank HS (dash-dotted purple), and our γ-BFF (solid red) methods. We include the RMSE for every method.

only one with a shaded region that indicates the error of the estimation.

An interesting result arises when we implement our method using the GH-BFF and the GL-BFF approaches for the approximation. Our method reproduces the GHFF and the GLFF classical Gaussian quadratures approximations in the limit, respectively, as expected from the discussion in Minka (2000) and Särkkä et al. (2016). The convergence takes place with a large enough feature parameter M. Detailed numerical results and an in-depth discussion on this topic are provided in the Supplementary Materials.

We next evaluate the performance of the four BFF approaches introduced earlier, examining their behavior across different values of the kernel length scale parameter σ. We approximate the squared exponential (SE) kernel on the interval $r \in [0, 1]$, fixing the kernel amplitude at $\alpha = 1.5$ and considering length scales

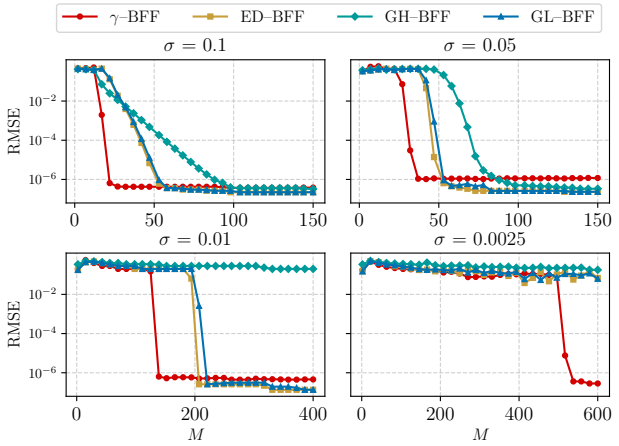


Figure 2: RMSE against the number of features M for our BFF method using the GH, GL, equidistant, and gamma approaches for approximation of the squared exponential kernel with different length scales $\sigma \in \{0.1, 0.05, 0.01, 0.0025\}$.

$\sigma \in \{0.1, 0.05, 0.01, 0.0025\}$. For all experiments, the BFF hyperparameters are held fixed at $\alpha_Q = 2S(0)$ and $\sigma_Q = 0.2$, with $\gamma = 1.6$ used exclusively for the γ–BFF variant.

The results presented in Figure 2 reveal that the γ–BFF approach achieves superior accuracy across a wide range of length scales compared to the ED–BFF, GH–BFF, and GL–BFF methods. Also, it converges earlier with respect to the other approaches, similar to the TQFF method proposed in Li et al. (2024). It is possible to compare the variance of the BFFs against the number of features to obtain an estimate of the number of features needed to get a feasible low-rank approximation. This result is shown in the Supplementary Materials.

Based on these findings, we adopt the γ–BFF approach in the following experiments reported in this work. It is important to note that the BFF framework is not limited to the specific node and kernel pair considered here. Alternative designs for the quadrature nodes and the auxiliary kernel k_Q could potentially lead to even stronger low-rank approximations, offering a promising direction for further research. More broadly, this flexibility illustrates how the BFF perspective provides a unifying lens through which different low-rank kernel approximations can be understood and systematically improved.

We evaluate the performance of different low-rank methods against the γ–BFF approach along different length scales. Similar to the previous experiment, we approximate the squared exponential (SE) kernel on the interval $r \in [0, 1]$, fixing the kernel

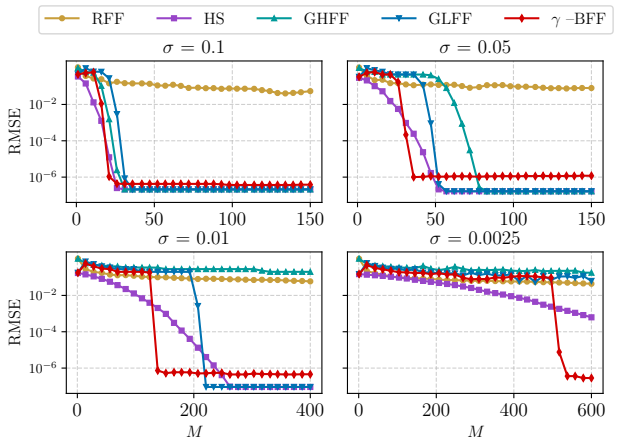


Figure 3: RMSE of kernel approximation methods (RFF, HS, GHFF, GLFF, BFF) as a function of the number of features for different length scales $\sigma \in \{0.1, 0.05, 0.01, 0.0025\}$.

amplitude at $\alpha = 1.5$ and considering length scales $\sigma \in \{0.1, 0.05, 0.01, 0.0025\}$. Figure 3 shows how the RMSE changes with the number of features for five kernel approximation methods (RFF, HS, GHFF, GLFF, and γ–BFF). At high length scales ($\sigma = 0.1, 0.05$), all the methods except the RFF quickly reduce RMSE as features increase, reaching single precision after about 20–70 features. RFF shows only small gains, even with many features. At lower length scales ($\sigma = 0.01$) and especially (0.0025), our BFF achieves the fastest error decay and very low RMSE with fewer features. The HS method improves gradually but does not achieve convergence for the shortest length scale; meanwhile, RFF remains the least accurate.

Using the BFF low rank kernel approximation, we can implement this kernel for different tasks such as regression, classification, or quadratures.

5 BFF FOR GAUSSIAN PROCESS REGRESSION

Similar to any Fourier features method, the BFF proposed in this paper can be used in different Gaussian process applications. In this section, we implement it for Gaussian process regression (GPR) to see its performance with respect to other methods. Details on the experiments are given in Supplementary Materials.

The low-rank kernel in Eq.(10) can be written as $k(r) = \hat{k} + \delta k$, where $\delta k \sim \mathcal{N}(0, \Sigma_k)$. Then, $\mathbf{k}_*$ and $\mathbf{K}$ are now written as $\hat{k}_* + \delta\mathbf{k}$ and $\hat{K} + \delta K$, where elements of δk and δK are joint Gaussian distributed according to the covariance function of the GP for $k(r)$. For big enough M features, the standard deviation of the BFF becomes small enough such that we can consider a first order perturbation $(\hat{K} + \delta K)^{-1} \approx \hat{K}^{-1} - \hat{K}^{-1}\delta K \hat{K}^{-1}$. Since the expected value of $\delta k = 0$, the mean of the low rank is not affected by the perturbation. Also, for small enough Σ, we can keep the covariance of the low-rank GPR such that, in a first approximation, it is not perturbed. More details on this are given in Supplementary Materials. As a result, the mean and variance for low-rank GPR using the BFF kernel are

$$\begin{aligned} \mathrm{E}[f_*] &= \hat{k}_*^\top (\hat{K} + \sigma_n^2 I)^{-1}\mathbf{y}, \\ \mathrm{V}[f_*] &= \hat{k}(x_*, x_*) - \hat{k}_*^\top (\hat{K} + \sigma_n^2 I)^{-1}\hat{k}_* \end{aligned} \tag{14}$$

Where $\hat{k}_*^\top = \hat{k}^\top(\mathbf{x}, \mathbf{x}_*)$ and $\hat{K}$ has components $\hat{K}_{ij} = \hat{k}(\mathbf{x}_i, \mathbf{x}_j)$ with $\mathbf{x}_i$ in the data set. Since the BFF low rank kernel $\hat{k}$ can be written in the form (11), it can be used in the GPR above to write it in the low rank form of Eq.(8). We gain a computational advantage by ensuring the matrix inversion in Eq.(8) has rank M instead of N, with $M \ll N$. This kernel reduction in rank minimizes the complexity associated with GPR.

We now compare different low-rank GPs for the regression of a noisy toy model with $N = 1000$ measurements and their performance against the complete GPR. Details of the experiment are given in Supplementary Materials. Figure 4 shows that all methods perform well in the region where the data is located. However, the RFF, GHFF, and GLFF methods lack accurate estimations of the GPR in regions far from the data. The HS and our γ–BFF methods outperform in this region, with the γ–BFF method having a slightly lower RMSE in both mean and variance of the regression with respect to the truth GPR.

We also implement different low-rank methods on a dataset that provides a series of measurements of head acceleration in a simulated motorcycle accident, used to test crash helmets (Venables and Ripley, 2002). Figure 5 shows how the considered low-rank GPs methods approximate the GPR using $M = 32$ features. Among these methods, γ-BFF demonstrates superior performance with the lowest root mean square error (RMSE) in both mean and variance, indicating its high accuracy and precision. The fit of γ-BFF closely aligns with the complete GPR for the dataset, alluding to its effectiveness in capturing the underlying trends. HS also performs moderately well, with uncertainty regions closer to the complete GPR compared to RFF, GHFF, and GLFF. The latter GHFF and GLFF models exhibit higher RMSE in both mean and variance with respect to RFF for this dataset. Overall, the γ–BFF method outperforms the low-rank methods considered, evidencing its robustness and reliability.

The evaluation of different low-rank Gaussian process

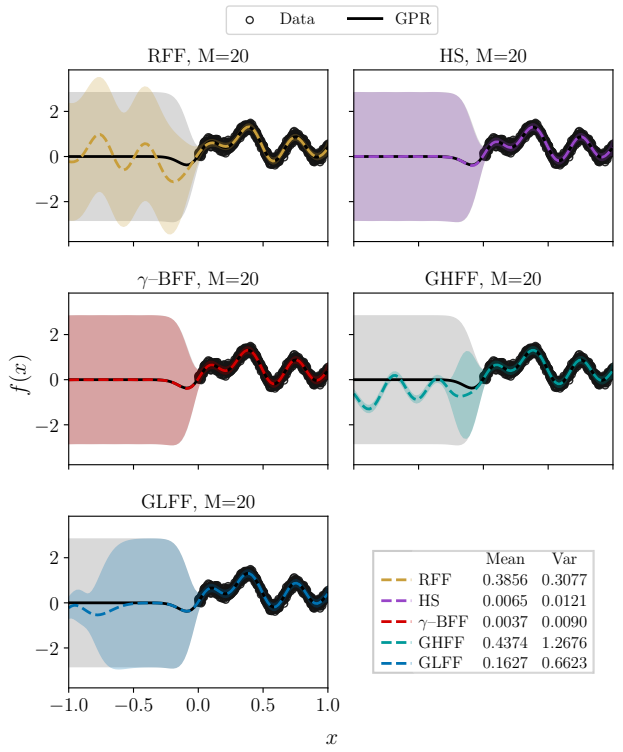


Figure 4: Comparison of low-rank GPs (colored dashed lines) against the complete GPR (black solid line) using different kernel approximations with $M = 20$. Methods include RFF, HS, γ–BFF, GHFF, and GLFF. The table reports RMSE for the predictive mean and variance, and the graphs include the predictive uncertainty in the shaded regions.

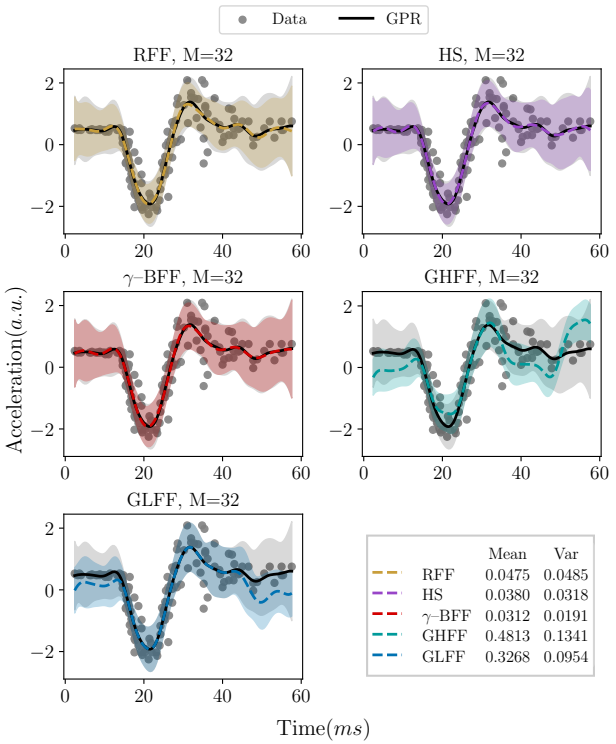


Figure 5: Comparison of low-rank GPs (colored dashed lines) against the complete GPR (black solid line) using different kernel approximations with $M = 32$ for motorcycle accident simulation. The data has time in the x axis and acceleration in the y axis. Methods include RFF, HS, γ–BFF, GHFF, and GLFF. The table reports RMSE for the predictive mean and variance, and the graphs include the predictive uncertainty in the shaded regions.

regression methods on a noisy toy model and a given dataset demonstrates the versatility and efficacy of the BFF approach proposed in this paper. These observations suggest that incorporating the BFF approach can substantially improve model robustness and predictive accuracy. Our BFF is not limited to the specific manner that was considered in this paper. This offers a promising direction for novel research to exploit the advantages of BFF for low-rank GPs.

We performed additional experiments in two-dimensional data for classification tasks. For this, we implemented low-rank methods on two binary classification tasks and a multi-class classification problem using the Iris dataset (Fisher, 1936). Details on the experiments are provided in Supplementary Materials. We also provide a repository for the method implementation[1]

[1]`https://github.com/cagalvisf/Bayesian_Fourier_Features`

6 CONCLUSION

This paper introduced a novel Bayesian Fourier feature approach for low-rank Gaussian processes. The method leverages Bayesian quadrature methods by solving Bochner's integral to construct a reduced-rank kernel approximation. Our BFF method delivers a principled uncertainty quantification for kernel reduced-rank approximations. We consider different approaches to construct the BFF. The GH-BFF and GL-BFF approximate the low-rank kernel obtained in the GHFF (Mutny and Krause, 2018) and GLFF (Shustin and Avron, 2022) methods, respectively, with a large enough number of features as expected from the discussion in Minka (2000) and Särkkä et al. (2016). The γ–BFF approach outperformed the classical Fourier features approaches considered here for kernel approximation. The reduced-rank kernel can be used for GP applications, leveraging the computational advantages of low-rank GPs. We apply our

model to a GPR problem and compare its performance with classical Fourier feature approaches, and observe BFF outperforms.

The proposed BFF method unifies many previously proposed low-rank methods, as they can be seen as instances of BFF for GP kernels. Examining alternative configurations for the quadrature nodes and the quadrature kernel k_Q offers the potential to achieve more robust low-rank approximations, thereby identifying a promising direction for future research. This flexibility portrays the ability of the BFF framework to provide a comprehensive perspective through which diverse low-rank kernel approximation methods can be understood and systematically improved.

As our method is a quadrature method for the Bochner integral, it only applies to stationary covariance functions. However, extension to non-stationary kernels, which have an integral representation as in Remes et al. (2017), could be considered in the future. A drawback of our method is the dimensionality scaling, as the number of nodes used to implement the quadrature rule increases exponentially with the dimensionality of the input data. This can be potentially overcome by using symmetric rules or sparse grids (Karvonen and Särkkä, 2018; Gerstner and Griebel, 1998).

Future work includes the implementation of the method for a broader set of spectral densities, delivering low-rank approximations of different kernels. The application of this low rank kernel is not limited to GPR, the kernel could be applied for GPQ or GP classification tasks. Besides exploring alternative configurations of the quadrature kernel k_Q, it is possible to optimize its hyperparameters such that the BFF delivers a low-rank kernel with better performance.

Acknowledgements

C.A. Galvis-Florez and Simo Särkkä thank the Research Council of Finland for funding this research.

References

N. I. Akhiezer and I. M. Glazman. *Theory of Linear Operators in Hilbert Space*. Dover Publications, 1993.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006. ISBN 978-0-387-31073-2.

Krzysztof Chalupka, Christopher K. I. Williams, and Iain Murray. A framework for evaluating approximation methods for Gaussian process regression. *Journal of Machine Learning Research*, 14(10):303–331, 2013. URL http://jmlr.org/papers/v14/chalupka13a.html.

Tri Dao, Christopher M De Sa, and Christopher Ré. Gaussian quadrature for kernel features. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/62f91ce9b820a491ee78c108636db089-Paper.pdf.

Volker L. Deringer, Albert P. Bartók, Noam Bernstein, David M. Wilkins, Michele Ceriotti, and Gábor Csányi. Gaussian process regression for materials and molecules. *Chemical Reviews*, 121 (16):10073–10141, 2021. doi: 10.1021/acs.chemrev.1c00022. URL https://doi.org/10.1021/acs.chemrev.1c00022. PMID: 34398616.

R. A. Fisher. Iris. UCI Machine Learning Repository, 1936. DOI: https://doi.org/10.24432/C56C76.

Thomas Gerstner and Michael Griebel. Numerical integration using sparse grids. *Numerical Algorithms*, 18(3):209–232, Jan 1998. ISSN 1572-9265. doi: 10.1023/A:1019129717644. URL https://doi.org/10.1023/A:1019129717644.

Philip Greengard and Michael O'Neil. Efficient reduced-rank methods for Gaussian processes with eigenfunction expansions. *Statistics and Computing*, 32(5):94, Oct 2022. ISSN 1573-1375. doi: 10.1007/s11222-022-10124-z. URL https://doi.org/10.1007/s11222-022-10124-z.

Oliver Hamelijnck, William Wilkinson, Niki Loppi, Arno Solin, and Theodoros Damoulas. Spatio-temporal variational Gaussian processes. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 23621–23633. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/c6b8c8d762da15fa8dbbdfb6baf9e260-Paper.pdf.

Philipp Hennig, Michael A Osborne, and Hans P Kersting. *Probabilistic Numerics: Computation as Machine Learning*. Cambridge University Press, 2022.

Toni Karvonen and Simo Särkkä. Classical quadrature rules via Gaussian processes. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2017. doi: 10.1109/MLSP.2017.8168195.

Toni Karvonen and Simo Särkkä. Fully symmetric kernel quadrature. *SIAM Journal on Scientific Computing*, 40(2):A697–A720, 2018. doi: 10.1137/17M1121779. URL https://doi.org/10.1137/17M1121779.

Toni Karvonen, Simo Särkkä, and Chris. J. Oates. Symmetry exploits for Bayesian cubature methods. *Statistics and Computing*, 29(6):1231–1248, Nov 2019. ISSN 1573-1375. doi: 10.1007/s11222-019-09896-8. URL https://doi.org/10.1007/s11222-019-09896-8.

Miguel Lázaro-Gredilla, Joaquin Quinonero-Candela, Carl Edward Rasmussen, and Aníbal R Figueiras-Vidal. Sparse spectrum Gaussian process regression. *Journal of Machine Learning Research*, 11: 1865–1881, 2010.

Kevin Li, Max Balakirsky, and Simon Mak. Trigonometric quadrature Fourier features for scalable Gaussian process regression. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 3484–3492. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/li24o.html.

Thomas P Minka. Deriving quadrature rules from Gaussian processes. Technical report, Carnegie Mellon University, 2000.

Mojmir Mutny and Andreas Krause. Efficient high dimensional Bayesian optimization with additivity and quadrature Fourier features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/4e5046fc8d6a97d18a5f54beaed54dea-Paper.pdf.

Anthony O'Hagan. Bayes–Hermite quadrature. *Journal of Statistical Planning and Inference*, 29(3):245–260, 1991.

Joaquin Quinonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

Carl Edward Rasmussen, Christopher K Williams, et al. Gaussian processes for machine learning, 2006.

Sami Remes, Markus Heinonen, and Samuel Kaski. Non-stationary spectral kernels. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/c65d7bd70fe3e5e3a2f3de681edc193d-Paper.pdf.

Simo Särkkä, Jouni Hartikainen, Lennart Svensson, and Fredrik Sandblom. Gaussian process quadratures in nonlinear sigma-point filtering and smoothing. In *17th International Conference on Information Fusion (FUSION)*, pages 1–8, 2014.

Simo Särkkä, Jouni Hartikainen, Lennart Svensson, and Fredrik Sandblom. On the relation between Gaussian process quadratures and sigma-point methods. *Journal of Advances in Information Fusion*, 11(1):31–46, June 2016. ISSN 1557-6418.

Paz Fink Shustin and Haim Avron. Gauss-legendre features for Gaussian process regression. *Journal of Machine Learning Research*, 23(92):1–47, 2022. URL http://jmlr.org/papers/v23/21-0030.html.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005.

Arno Solin and Simo Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, 30(2):419–446, 2020.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In David van Dyk and Max Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR. URL https://proceedings.mlr.press/v5/titsias09a.html.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.

Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000. URL https://proceedings.neurips.cc/paper_files/paper/2000/file/19de10adbaa1b2ee13f77f679fa1483a-Paper.pdf.

Max A. Woodbury. *Inverting modified matrices*. Princeton University, Princeton, NJ, 1950. Statistical Research Group, Memo. Rep. no. 42,.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes**
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Yes**
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Yes**

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Yes**
   (b) Complete proofs of all theoretical results. **Not Applicable**
   (c) Clear explanations of any assumptions. **Yes**

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Yes**
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes**
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes**
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **Yes**

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. **Yes**
   (b) The license information of the assets, if applicable. **Not Applicable**
   (c) New assets either in the supplemental material or as a URL, if applicable. **Not Applicable**
   (d) Information about consent from data providers/curators. **Not Applicable**
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not Applicable**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. **Not Applicable**
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not Applicable**
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not Applicable**

Supplementary Materials

A MATHEMATICAL EQUIVALENCE OF QUADRATURE UNDER CHANGE OF INTEGRAND

This appendix clarifies the Bayesian quadrature formulation of our BFF method for stationary kernels, and why it is sufficient to consider the case $\mu(\omega) = 1$. Leveraging Bochner's representation of a stationary kernel as an integral over its spectral density, we show that considering either $S(\omega)\cos(\omega r)$ a GP with a uniform density $\mu(\omega) = 1$, or modeling $\cos(\omega r)$ as a GP with a weighted density $\mu(\omega) = S(\omega)$, produce identical integral estimators when the kernel function is chosen consistently. Therefore, the choice of $\mu(\omega) = 1$ in the paper is without a loss of generality. We derive this equivalence in a general setting for integrals of the form $\int f(\omega)\mu(\omega)d\omega$, compare the corresponding GP approximations, and demonstrate that the resulting quadrature rules coincide. This establishes that the two choices of weighting between the integrand and the density are equivalent for BFF under the correct kernel choice.

In Section 3, we introduced a Bayesian quadrature approach to approximate stationary kernels that, using Bochner's theorem, can be written as

$$k(r) = \frac{1}{2\pi}\int_{\mathbb{R}} S(\omega)\cos(\omega r)\, d\omega.$$

It is possible (if we discard the constant) to consider either $S(\omega)\cos(\omega r)$ as a GP with density function $\mu(\omega) = 1$ or $\cos(\omega r)$ as a GP with density $\mu(\omega) = S(\omega)$. Here, we show that both choices are equivalent under the correct kernel choice. For this, we can consider a more general integration

$$\mathcal{I} = \int f(\omega)\mu(\omega)d\omega.$$

Using evaluation points $\Xi = \{\xi_i\}_{i=0}^{M}$ we can propose a GP approximation of either $f(\omega)$ or $f(\omega)\mu(\omega)$. In the first case, we would get an approximation for $f(\omega)$ using a GP with covariance $k_1(\omega, \omega')$ that takes the form

$$f(\omega) \approx f_1(\omega) = k_1(\omega, \Xi)K_1(\Xi, \Xi)^{-1} f(\Xi),$$

where $f(\Xi)$ is a vector with components $f(\xi_i)$, the matrix $K_1(\Xi, \Xi)$ has components $k_1(\xi_i, \xi_j)$ and the vector $k_1(\omega, \Xi)$ has components $k_1(\omega, \xi_i)$. This can be used to build an approximation for the integral $\mathcal{I} \approx \mathcal{I}_1$, which gives

$$\begin{aligned}
\mathcal{I} \approx \mathcal{I}_1 &= \int f_1(\omega)\mu(\omega)\, d\omega \\
&= \int k_1(\omega, \Xi)K_1(\Xi, \Xi)^{-1} f(\Xi)\mu(\omega)\, d\omega \\
&= \left[\int k_1(\omega, \Xi)\mu(\omega)\, d\omega\right] K_1(\Xi, \Xi)^{-1} f(\Xi).
\end{aligned}$$

Alternatively, we can approximate $f(\xi)\mu(\xi)$ as a GP characterized by a covariance function $k_2(\omega, \omega')$. This gives us

$$f(\omega)\mu(\omega) \approx f_2(\omega) = k_2(\omega, \Xi)K_2(\Xi, \Xi)^{-1}\,\mathrm{diag}(\mu(\Xi))f(\Xi),$$

where the vectors and matrices have similar components as in the previous case. This delivers the approximation

for the integral

$$\begin{aligned}\mathcal{I} \approx \mathcal{I}_2 &= \int f_2(\omega) d\omega \\ &= \int k_2(\omega, \Xi) K_2(\Xi, \Xi)^{-1} \operatorname{diag}(\mu(\Xi)) f(\Xi) d\omega \\ &= \left[\int k_2(\omega, \Xi)\, d\omega\right] K_2(\Xi, \Xi)^{-1} \operatorname{diag}(\mu(\Xi)) f(\Xi).\end{aligned}$$

Let us now define

$$k_2(\omega, \omega') = \mu(\omega) k_1(\omega, \omega') \mu(\omega').$$

Then, replacing in the second approximation, we would get

$$I_2 = \left[\int \mu(\omega) k_1(\omega, \Xi) \operatorname{diag}(\mu(\Xi)) d\omega\right] \left[\operatorname{diag}(\mu(\Xi)) K_1(\Xi, \Xi) \operatorname{diag}(\mu(\Xi))\right]^{-1} \operatorname{diag}(\mu(\Xi)) f(\Xi).$$

Using the matrix inversion property $(AB)^{-1} = B^{-1}A^{-1}$, we get

$$I_2 = \left[\int k_1(\omega, \Xi) \mu(\omega)\, d\omega\right] K_1(\Xi, \Xi)^{-1} f(\Xi),$$

which gives

$$I_2 = I_1.$$

This means that, considering $f(\omega)$ or $f(\omega)\mu(\omega)$ as a GP, the Bayesian quadrature obtained for the integral is the same. For our BFF method, choosing $S(\omega)\cos(\omega r)$ as a GP with density function $\mu(\omega) = 1$ or $\cos(\omega r)$ as a GP with density $\mu(\omega) = S(\omega)$ is equivalent.

B GAUSSIAN QUADRATURE NODES FOR BFF

In this appendix, we present results for the GH-BFF and GL-BFF methods applied to the squared exponential kernel and analyze their convergence toward their classical counterparts, GHFF and GLFF, respectively, as the number of features increases. We discuss this in the context of Gaussian process quadratures, connecting to previous work by Minka (2000) and Särkkä et al. (2016). These results clarify how our BFF approximations recover classical quadratures in the limit of a large enough feature parameter.

Section 4 provides details on the implementation of the BFF method for a one-dimensional squared exponential kernel. The construction of the BFF method can vary based on the selection of nodes, integration limits, and quadrature kernel. For this analysis, we propose a quadrature kernel defined by $k_Q = \alpha_Q \exp\left(-(\omega - \omega')^2 / 2\sigma_Q^2\right)$ with $\alpha_Q = 2S(0)$ and $\sigma_Q = 0.4$. We explore four different approaches that vary in terms of the domain of integration and choice of nodes. Among these approaches, the GH-BFF and GL-BFF methods are characterized by using the roots of Hermite and Legendre polynomials as nodes for the BFF method.

Figures 6 and 7 show the results of the GH-BFF and GL-BFF methods, respectively, and their convergence to their classical counterparts, the GHFF and GLFF.

In the results, we observe how the BFF features converge to the low-rank kernel approximation of the method, whose nodes have been utilized. When the GH nodes are used, we observe how our BFF method approaches the GHFF kernel approximation, as well as the GLFF. These results can be understood from the analysis made by Minka (2000) and Särkkä et al. (2016), in which the squared exponential kernel can be seen as

$$\begin{aligned}k_Q(\omega, \omega') &\propto e^{-\frac{(\omega - \omega')^2}{2\sigma_Q^2}} \\ &= e^{-\frac{\omega^2}{2\sigma_Q^2}} e^{\frac{\omega\omega'}{\sigma_Q^2}} e^{-\frac{\omega'^2}{2\sigma_Q^2}} \\ &= e^{-\frac{\omega^2}{2\sigma_Q^2}} \left(\sum_i \frac{1}{i! \sigma_Q^{2i}} \omega^i \omega'^i\right) e^{-\frac{\omega'^2}{2\sigma_Q^2}} \\ &= \left(\sum_i \frac{1}{i! \sigma_Q^{2i}} \omega^i e^{-\frac{\omega^2}{2\sigma_Q^2}}\right) \left(\sum_i \frac{1}{i! \sigma_Q^{2i}} \omega'^i e^{-\frac{\omega'^2}{2\sigma_Q^2}}\right).\end{aligned}$$

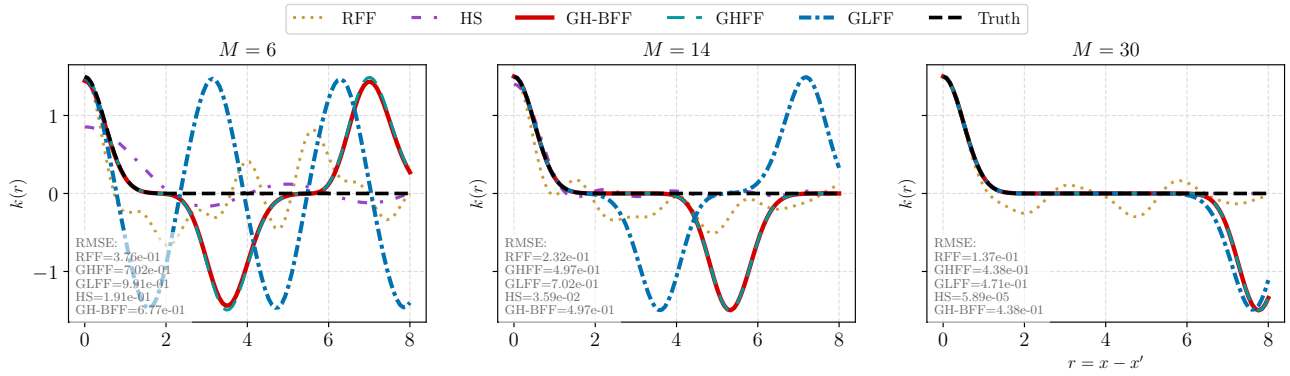


Figure 6: Approximation of the squared exponential kernel (dashed black) using RFF (dotted gold), GHFF (dashed teal), GLFF(dash-dotted blue), low-rank HS (dash-dotted purple), and our GH-BFF (solid red) methods. We include a measure of the RMSE for every approximation.

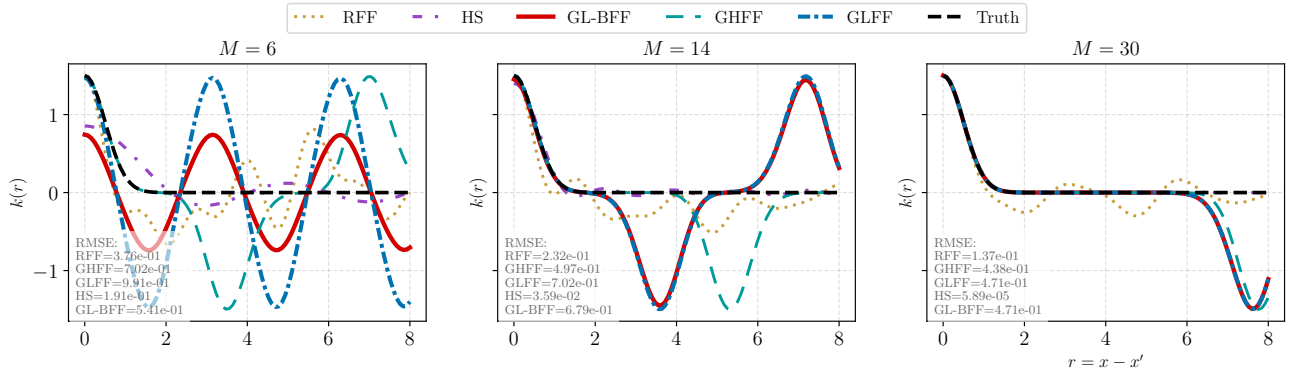


Figure 7: Approximation of the squared exponential kernel (dashed black) using RFF (dotted gold), GHFF (dashed teal), GLFF(dash-dotted blue), low-rank HS (dash-dotted purple), and our GL-BFF (solid red) methods. We include a measure of the RMSE for every approximation.

The terms in brackets correspond to damped polynomials that in the limit of $\sigma_Q \to \infty$ converge to a polynomial basis. This behavior of the SE kernel reproducing classical quadratures was already pointed out by Minka (2000) and Särkkä et al. (2016). Previous papers have spotted this relation between the squared exponential kernel converging to classical quadrature rules in the limit; however, a rigorous proof of this behavior does not yet seem to appear in the literature.

C VARIANCE OF THE BFF AGAINST NUMBER OF FEATURES

In this appendix, we discuss how the variance estimates produced by the BFF method can be used to determine the number of features employed in a low-rank Gaussian process approximation.

In Section 4, we implemented the BFF method for low rank approximation using the γ, equidistant, Gauss-Hermite, and Gauss-Legendre nodes. For each of these configurations, Eq. (10) provides a variance that quantifies the uncertainty of the approximation for a finite number of features.

Figure 8 shows how the variance of the BFF changes as the number of features increases for the different node configurations. In every case, the variance decreases with respect to the number of features as expected, indicating an improved accuracy of the low-rank approximation. By fixing a variance tolerance for the kernel approximation, it would be possible to select a suitable number of features for the method. This provides a criterion to select the number of features for the low-rank approximation, independent of the input data.

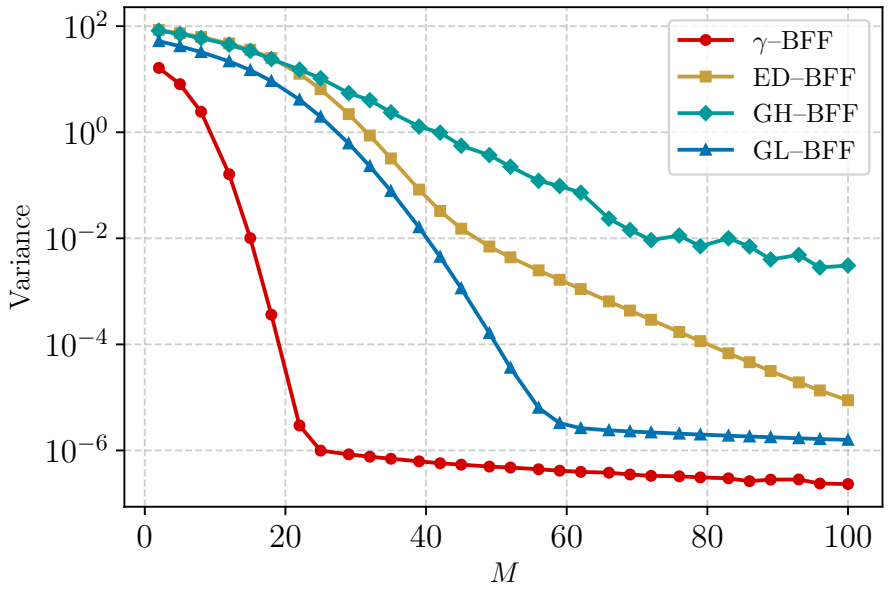


Figure 8: The plot shows the variance obtained from the BFF method against different numbers of features for the 4 node types (γ, GH, GL, ED). We can choose a variance tolerance for the kernel approximation that would deliver the number of features needed for the low-rank kernel.

D INSIGHTS IN THE APPLICATION OF BFF TO GPR

This appendix discusses the mathematical details of employing our BFF kernel approximation for a GPR problem. We demonstrate how the approximation is implemented as other low-rank kernel approximations, especially when the feature count M becomes sufficiently large. We show in detail how the posterior mean and variance of our low-rank approximation can be effectively implemented in GPR.

In Section 5, we applied our BFF method for GPR. For this we recall the kernel approximation $k(r) = \hat{k} + \delta k$, where $\delta\mathbf{k}$ follows a Gaussian distribution with mean 0 and covariance Σ_k, denoted as $\delta k \sim \mathcal{N}(0, \Sigma_k)$. Then, $\mathbf{k}*$ and $\mathbf{K}$ can be expressed as $\hat{k} * + \delta\mathbf{k}$ and $\hat{K} + \delta K$, respectively. The elements of $\delta\mathbf{k}$ and δK are jointly Gaussian, in accordance with the covariance function of the Gaussian process for $k(r)$. When the number of features M is sufficiently large, the standard deviation of the BFF becomes sufficiently small, allowing the use of a first-order perturbation approximation $(\hat{K} + \delta K)^{-1} \approx \hat{K}^{-1} - \hat{K}^{-1}\delta K\hat{K}^{-1}$. Applying this kernel to GPR in Eq. (1), considering the case of $\sigma_n = 0$ to simplify the algebra, allows us to rewrite the mean

$$\begin{aligned}
\mathbb{E}[f_*] &= \left(\hat{k}_* + \delta\mathbf{k}\right)\left(\hat{K} + \delta K\right)^{-1}\mathbf{y} \\
&\approx \left(\hat{k}_* + \delta\mathbf{k}\right)\left(\hat{K}^{-1} - \hat{K}^{-1}\delta K\hat{K}^{-1}\right)\mathbf{y} \\
&= \hat{\mathbf{k}}_*\hat{K}^{-1}\mathbf{y} - \hat{\mathbf{k}}_*\hat{K}^{-1}\delta K\hat{K}^{-1}\mathbf{y} + \delta\mathbf{k}\hat{K}^{-1}\mathbf{y} - \delta\mathbf{k}\hat{K}^{-1}\delta K\hat{K}^{-1}\mathbf{y}
\end{aligned}$$

Since the expected value $\mathbb{E}[\delta k] = 0$, then the terms with δk cancel out, leaving what can be recognized as the usual GPR mean, but using the kernel approximation. Including the noise parameter, we get

$$\mathbb{E}[f_*] = \hat{\mathbf{k}}_*\left(\hat{K} + \sigma_n^2 I\right)^{-1}\mathbf{y}.$$

Now we use the kernel approximation for the variance of GPR. As a result, we get

$$\begin{aligned}
\mathbb{V}[f_*] &= \left(\hat{k}(\mathbf{x}_*, \mathbf{x}_*) + \delta k\right) - \left(\hat{k}_* + \delta\mathbf{k}\right)^\top \left(\hat{K} + \delta K\right)^{-1} \left(\hat{k}_* + \delta\mathbf{k}\right) \\
&\approx \left(\hat{k}(\mathbf{x}_*, \mathbf{x}_*) + \delta k\right) - \left(\hat{k}_* + \delta\mathbf{k}\right)^\top \left(\hat{K}^{-1} - \hat{K}^{-1}\delta K\hat{K}^{-1}\right) \left(\hat{k}_* + \delta\mathbf{k}\right) \\
&= \left(\hat{k}(\mathbf{x}_*, \mathbf{x}_*) + \delta k\right) - \hat{k}_*^\top\hat{K}^{-1}\hat{k}_* - \hat{k}_*^\top\hat{K}^{-1}\delta k + \hat{k}_*^\top\hat{K}^{-1}\delta K\hat{K}^{-1}\hat{k}_* + \hat{k}_*^\top\hat{K}^{-1}\delta K\hat{K}^{-1}\delta\mathbf{k} \\
&\quad - \delta\mathbf{k}^\top\hat{K}^{-1}\hat{k}_* - \delta\mathbf{k}^\top\hat{K}^{-1}\delta\mathbf{k} + \delta\mathbf{k}^\top\hat{K}^{-1}\delta K\hat{K}^{-1}\hat{k}_* + \delta\mathbf{k}^\top\hat{K}^{-1}\delta K\hat{K}^{-1}\delta\mathbf{k}.
\end{aligned}$$

Discarding the terms of order superior to $O(\delta k^2)$, we get

$$\mathbb{V}[f_*] = \hat{k}(\mathbf{x}_*, \mathbf{x}_*) - \hat{k}_*^\top\hat{K}^{-1}\hat{k}_* + \delta k - \hat{k}_*^\top\hat{K}^{-1}\delta\mathbf{k} + \hat{k}_*^\top\hat{K}^{-1}\delta K\hat{K}^{-1}\hat{k}_* - \delta\mathbf{k}^\top\hat{K}^{-1}\hat{k}_*.$$

For increasing M, the terms $\delta\mathbf{k} \ll \hat{k}_*$, this is supported by results in Figure 1. The variance above adds a contribution that propagates to the predictions, however, the term $\hat{k}_*^\top\hat{K}^{-1}\delta K\hat{K}^{-1}\hat{k}_*$ increases the complexity, disrupting the advantage of using a low-rank approximation, so we discard this term. Then, in our applications, we identify the usual term for the variance in GPR, and adding the noise term, we get

$$\mathbb{V}[f_*] = \hat{k}(\mathbf{x}_*, \mathbf{x}_*) - \hat{k}_*^\top \left(\hat{K} + \sigma_n^2 I\right)^{-1} \hat{k}_*.$$

This means that, in this first approach, we consider the low-rank kernel approximation $\hat{k}(r)$ of $k(r)$ as our kernel. This allows us to also write the GPR equations in their low-rank form in Eq. (8) similarly to well-known reduced-rank kernel approximations.

E DETAILS ON THE EXPERIMENTS FOR GPR

This appendix provides detailed descriptions of the experimental setups used to compare the performance of our γ–BFF method with popular low-rank approximation methods for GPR. In the first subsection, we use a synthetic noisy toy model, detailing the data generation process, kernel configurations, and the details for the implementations of low-rank methods. The second subsection includes a real-world dataset measuring head acceleration during a simulated motorcycle accident. We describe how the data were obtained, normalized, and used to test the methods under this given dataset.

E.1 Noisy toy model

In Section 5, we implemented our γ–BFF method for GPR along with different low-rank methods. In the first application, we have $N = 1000$ samples of the function

$$y = \frac{1}{2}\sin(6\pi x) + e^{-50(x-0.3)^3} + \frac{1}{2}x + \epsilon,$$

where $\epsilon \sim \mathcal{N}(0, 0.1^2)$, for x in the interval $[0, 1]$. The low-rank GPR is implemented for x_* in the interval $\in [-1, 1]$. The kernel used for the regression is the SE kernel $k(x, x') = \alpha\exp(-(x - x')^2/2\sigma^2)$ with $\alpha = 2$ and $\sigma = 0.1$. All the low-rank methods use $M = 20$ features. The HS method is implemented with a Fourier basis $\sin(\lambda_j(x + L)/\sqrt{L})$ with $L = 2$, for details on this method, please see Solin and Särkkä (2020). The GLFF method is implemented by truncating the integral in the interval $[-4.5, 4.5]$. Our BFF method is implemented using the SE kernel $k_Q(\omega, \omega') = \alpha_Q\exp(-(\omega - \omega')^2/2\sigma_Q^2)$ with $\alpha_Q = 2S(0)$, $\sigma_Q = 0.7$ and $\gamma = 1.05$

E.2 Head acceleration in a simulated motorcycle accident

We used a dataset that provides a series of measurements of head acceleration in a simulated motorcycle accident, used to test crash helmets (Venables and Ripley, 2002). In Python, we can get this dataset by `import statsmodels.api.get_rdataset("mcycle", package="MASS").data`. We used the time and acceleration of the head as our x and y variables. Then we normalize the acceleration data to have zero mean and unit standard deviation.

The low-rank GPR is implemented in the interval $x_* \in [0, 60]$. The kernel used for the regression is the SE kernel $k(x, x') = \alpha \exp(-(x - x')^2/2\sigma^2)$ with $\alpha = 4$ and $\sigma = 3$. All the low-rank methods use $M = 32$ features. The HS method is implemented with a Fourier basis $\sin(\lambda_j(x + L)/\sqrt{L})$ with $L = 60$. The GLFF method is implemented by truncating the integral in the interval $[-4.5, 4.5]$. Our BFF method is implemented using the SE kernel $k_Q(\omega, \omega') = \alpha_Q \exp(-(\omega - \omega')^2/2\sigma_Q^2)$ with $\alpha_Q = 2S(0)$, $\sigma_Q = 1$ and $\gamma = 0.85$.

F ADDITIONAL EXPERIMENTS FOR 2D DATA

This appendix provides additional experiments for the implementation of our γ–BFF method applied for data classification, and compares its performance with popular low-rank Gaussian process approximation methods. We follow the classification setup in Rasmussen et al. (2006) for labeled data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$.

Given a test point $\mathbf{x}_*$ GPR provides a posterior distribution with mean μ_* and variance σ_*^2 given the data $\mathcal{D}$. The probability that $\mathbf{x}_*$ is assigned label $y_* = 1$ is then

$$p(y_* = 1|\mathcal{D}) = \Phi\left(\frac{\mu_*}{\sqrt{1 + \sigma_*^2}}\right), \tag{15}$$

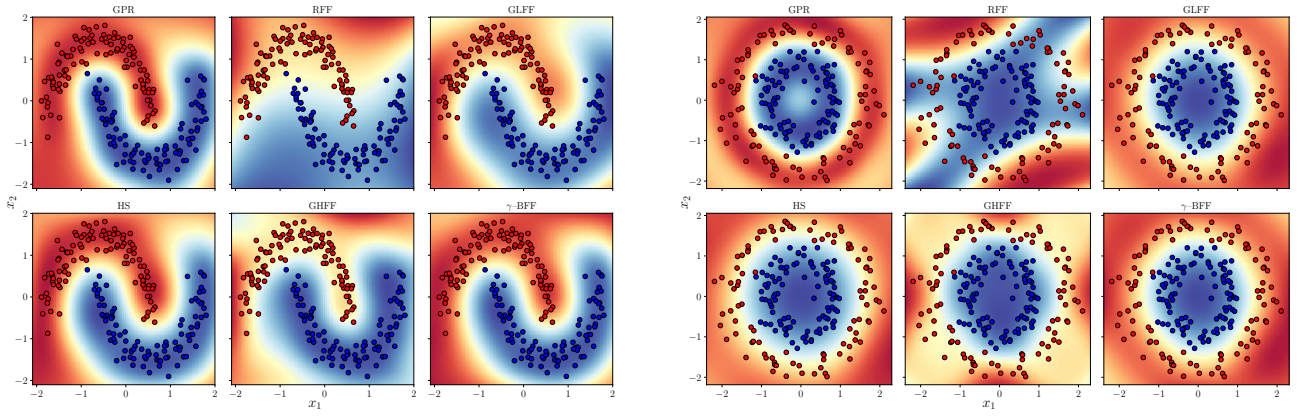
where we use a probit activation function of the form (Bishop, 2006)

$$\Phi(x) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right). \tag{16}$$

This classification procedure is used for binary and multi-class classification problems. In the case of multi-class tasks, like the ones in the Iris dataset, we used a one-versus-all strategy. In this approach, a separate binary classifier is trained for each class, where samples belonging to the target class are distinguished from all others. The final class assignment is then obtained by combining the probabilistic outputs of these individual classifiers.

F.1 Binary classification

We perform Gaussian process classification on the make-moons and make-circles datasets, which are commonly used for binary classification. For this classification task, we aim to use the minimum number of features that still achieves a satisfactory performance. The results are shown in Figure 9.



(a) Make moons dataset ($M = 6$). (b) Make circles dataset ($M = 4$).

Figure 9: Comparison of low-rank Gaussian process classifiers against the full GPR using different kernel approximations. Methods include RFF, HS, γ–BFF, GHFF, and GLFF, evaluated on binary datasets.

F.2 Iris dataset

The classification framework described above is extended to the multi-class setting by adopting a one-versus-all strategy as in the beginning of Sec F.

This multi-class classification scheme is evaluated using the UCI Iris dataset (Fisher, 1936), where we classify the three flower species Setosa, Versicolor, and Virginica based on their sepal length and width. The resulting classification regions are visualized in Figure 10.

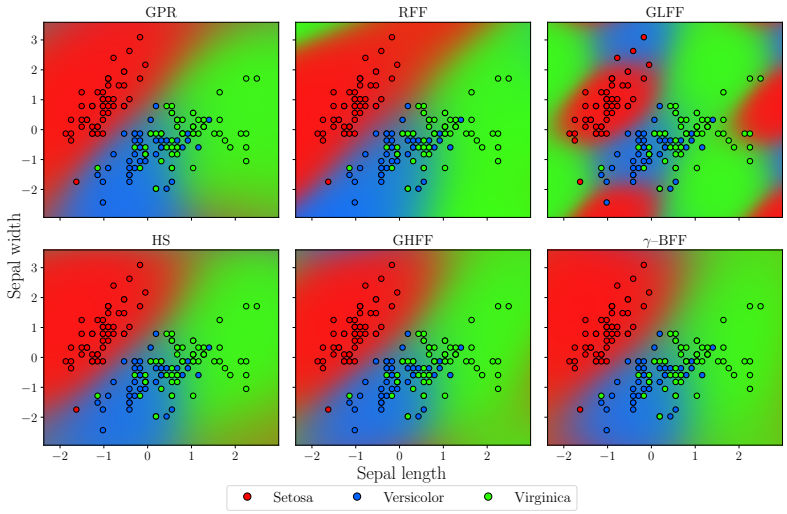


Figure 10: Comparison of low-rank GPs against the complete GPR using different kernel approximations with $M = 3$ for the UCI iris dataset. Methods include RFF, HS, γ–BFF, GHFF, and GLFF.

Table 1: RMSE for the predicted labels in classification of the approximation methods with respect to the GPR for the make moons, make circles, and the Iris dataset

Method	**Moons**	**Circles**	**Iris**
RFF	0.0813	0.2031	0.0379
GLFF	0.0654	0.0808	0.0743
GHFF	0.0829	0.0830	0.0181
HS	0.0365	0.1199	0.0119
BFF	0.0347	0.0789	0.0195

To compare the accuracy of the proposed low-rank approximation methods, we evaluate their performance by computing the RMSE with respect to the reference Gaussian process classifier. The RMSE is calculated between the predicted class probabilities obtained from each approximation method and those produced by the full GPR model. This comparison is performed for both the binary and multi-class classification tasks. The resulting RMSE values are shown in Table 1.