Trigonometric Quadrature Fourier Features for Scalable Gaussian Process Regression

Kevin Li
Duke University

Max Balakirsky
Duke University

Simon Mak
Duke University

Abstract

Fourier feature approximations have been successfully applied in the literature for scalable Gaussian Process (GP) regression. In particular, Quadrature Fourier Features (QFF) derived from Gaussian quadrature rules have gained popularity in recent years due to their improved approximation accuracy and better calibrated uncertainty estimates compared to Random Fourier Feature (RFF) methods. However, a key limitation of QFF is that its performance can suffer from well-known pathologies related to highly oscillatory quadrature, resulting in mediocre approximation with limited features. We address this critical issue via a new Trigonometric Quadrature Fourier Feature (TQFF) method, which uses a novel non-Gaussian quadrature rule specifically tailored for the desired Fourier transform. We derive an exact quadrature rule for TQFF, along with kernel approximation error bounds for the resulting feature map. We then demonstrate the improved performance of our method over RFF and Gaussian QFF in a suite of numerical experiments and applications, and show the TQFF enjoys accurate GP approximations over a broad range of length-scales using fewer features.

1 INTRODUCTION

Gaussian Processes (GPs) (Rasmussen and Williams, 2005) are a popular class of Bayesian non-parametric models. Unfortunately, for large sample sizes $n \gg 1000$, the $\mathcal{O}(n^3)$ cost for GP training and prediction can be prohibitive in applications. There has been much work on addressing this critical issue, including inducing points (Titsias, 2009; Hensman et al., 2013; Snelson and Ghahramani, 2005), nearest-neighbor approximations (Wu et al., 2022; Cao et al., 2023; Katzfuss and Guinness, 2021), iterative numerical methods (Gardner et al., 2018; Lin et al., 2023; Wang et al., 2019), and divide-and-conquer approaches (Deisenroth and Ng, 2015; Zhang and Williamson, 2019).

Our paper will focus the use of Fourier feature approximations (Rahimi and Recht, 2007), which have shown promise in recent work. The key idea is to construct a low-rank approximation of the covariance matrix for a stationary GP, using a finite set of Fourier features dervied from the kernel's spectral density. Fourier approximations have three key advantages for GP regression: they allow for kernel covariance approximation error bounds, reduce the non-parametric regression problem to linear regression, and exploit the spectral representation of covariance kernels. As such, such approximations are increasingly popular in broad applications, including generalized Bayesian quadrature (Warren et al., 2022), deep GPs (Cutajar et al., 2017), latent variable models (Zhang et al., 2023), differential privacy (Dubey, 2021; Dai et al., 2021), federated learning (Dai et al., 2020), Bayesian optimization, (Deng et al., 2022; Mutny and Krause, 2018), and spatial statistics (Ton et al., 2018).

For GPs, there has been two main directions for Fourier feature approximation. The first direction, Random Fourier Features (RFF; Rahimi and Recht, 2007), uses Monte Carlo sampling to generate features. The integration of RFF for GP regression is easy-to-implement and scales nicely to high dimensions. However, RFF methods are known to suffer from a phenomenon called *variance starvation*, which can lead to poorly calibrated uncertainty estimates and erratic predictive mean behavior (Wilson et al., 2020, 2021; Mutny and Krause, 2018; Wang et al., 2018).

The second direction, Quadrature Fourier Features (QFF; Dao et al., 2017; Mutny and Krause, 2018; Shustin and Avron, 2022), aims to alleviate vari-

Proceedings of the 27$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

ance starvation via a deterministic Gaussian quadrature rule for the Fourier transform integral. This has been successfully applied for lower-dimensional GP applications, including Bayesian optimization (Dai et al., 2021; Dubey, 2021; Mutny and Krause, 2018; Ray Chowdhury and Gopalan, 2019), robust inference (Qing et al., 2022), spatial-temporal data (Shustin and Avron, 2022), and derivative modeling (Angelis et al., 2020). Compared to RFF, the deterministic quadrature rules in QFF permit quicker error decay and can thus avoid variance starvation. In practice, however, achieving this improved performance over RFF can require an undesirably large number of features, particularly with small length-scales for the underlying GP (Mutny and Krause, 2018; Shustin and Avron, 2022).

This paper proposes a new Trigonometric Quadrature Fourier Features (TQFF) that addresses the aforementioned limitations of existing RFF and QFF methods. We show that the use of Gaussian quadrature rules in QFF (which rely on polynomial interpolants) can lead to poor performance with small length-scales when approximating the highly oscillatory Fourier transform. Motivated by this, the TQFF uses a novel quadrature rule that relies on a *trigonometric* interpolant, tailored specifically for the desired Fourier transform. In doing so, we show empirically that the TQFF enjoys accurate GP approximations over a broad range of length-scales using fewer features. We also provide a code implementation[1].

2 BACKGROUND

2.1 Gaussian Process Regression

A Gaussian process $f(\cdot)$ is a stochastic process for which its evaluation on any finite subset of inputs follows a multivariate Gaussian distribution. Here, we assume the standard regression set-up, with observed data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. The standard GP regression framework then follows:

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2),$$

where $f(\cdot) \sim \mathcal{GP}(0, k_{\mathbf{\Theta}}(\cdot,\cdot))$ follows a zero-mean GP with kernel $k_{\mathbf{\Theta}}$. Here, $\mathbf{\Theta}$ consists of all kernel hyperparameters, including length-scale and scale parameters. Training of these kernel hyperparameters $\mathbf{\Theta}$ and the noise variance σ^2 can proceed via maximizing the following log-marginal likelihood of $\mathbf{y} = (y_i)_{i=1}^n$:

$$\log p(\mathbf{y}|\mathbf{X}, \mathbf{\Theta}) = \log \phi(\mathbf{y}; \mathbf{0}, \mathbf{K_{XX}} + \mathbf{I}_{n\times n}\sigma^2),$$

where $\phi(\cdot; \boldsymbol{\mu}, \Sigma)$ is the multivariate Gaussian density. Here, $\mathbf{K_{XX}} \in \mathbb{R}^{n\times n}$ is the covariance matrix with (i,k)-th entry $k_{\mathbf{\Theta}}(\mathbf{x}_i, \mathbf{x}_k)$. Model inference and prediction thus requires the inversion of a $n \times n$ matrix, which requires $\mathcal{O}(n^3)$ operations and can thus be prohibitive for large $n \gg 1000$.

[1] https://github.com/kevinli1324/TQFF

2.2 Gaussian Quadrature

We provide a brief review of classical Gaussian quadrature; for details, see Chapter 7 in Conte and Boor (1980). Gaussian quadrature approximates integrals of the form:

$$\int_a^b p(\omega)h(\omega)d\omega, \quad -\infty \le a < b \le \infty,$$

where $p(\omega)$ is the *weight* function and $h(\omega)$ is the *integrand*. Different weight functions give rise to different quadrature rules. Gaussian quadrature makes the approximation $h(\omega) \approx P_{L-1}(\omega)$ where $P_{L-1}(\omega)$ is an $L-1$ degree polynomial interpolating $h(\omega)$ at L quadrature nodes $\{\omega_l\}_{l=1}^L$, $\omega_l \in [a,b]$. With this approximation, the quadrature rule becomes:

$$\int_a^b p(\omega)h(\omega)d\omega \approx \sum_{l=1}^L a_l h(\omega_l) := Q_L(h), \qquad (1)$$

where $a_l = \int_a^b h(\omega)t_l(\omega)d\omega$ and $t_l(\omega)$ are the $L-1$-degree Lagrange interpolating polynomials (Conte and Boor, 1980). This quadrature rule thus requires the set of quadrature nodes $\{\omega_l\}_{l=1}^L$ and quadrature weights $\{a_l\}_{l=1}^L$. Gaussian quadrature rules select these nodes such that Equation (1) is exact for polynomial $h(\omega)$ of degree up to $2L-1$. However, the accuracy of such quadrature rules (and polynomially-exact quadrature rules in general) depends on how well the polynomial interpolant approximates the integrand $h(\omega)$.

The above 1-dimensional quadrature rules can directly be extended for multiple dimensions via tensor products; details of this in Appendix 7. We do note that, with tensor product rules, the number of nodes grows exponentially with dimensions, which can limit such approaches to problems in low dimensions or with low-dimensional structure.

2.3 Fourier Feature Approximation

We now briefly review the general Fourier feature approximation approach. Using Bochner's theorem any stationary covariance function $k_{\mathbf{\Theta}}(\mathbf{x}, \mathbf{x}') = k_{\mathbf{\Theta}}(\mathbf{x} - \mathbf{x}')$ can be represented as the Fourier transform of a non-negative measure $p_{\mathbf{\Theta}}(\omega)$ (Rasmussen and Williams (2005) Section 4.2.1):

$$\begin{aligned} k_{\mathbf{\Theta}}(\mathbf{x} - \mathbf{x}') &= \int p_{\mathbf{\Theta}}(\boldsymbol{\omega}) \exp(i\boldsymbol{\omega}^T(\mathbf{x} - \mathbf{x}'))d\boldsymbol{\omega} \qquad (2) \\ &= \int p_{\mathbf{\Theta}}(\boldsymbol{\omega}) \cos(\boldsymbol{\omega}^T(\mathbf{x} - \mathbf{x}'))d\boldsymbol{\omega}, \end{aligned}$$

where the last line assumes both data and kernel are real-valued. Fourier feature methods then use the following finite feature approximation:

$$k_{\boldsymbol{\Theta}}(\mathbf{x}, \mathbf{x}') \approx \sum_{s=1}^{S} a_s \cos(\boldsymbol{\omega}_s^T(\mathbf{x} - \mathbf{x}')) = \boldsymbol{\Phi}(\mathbf{x})^T \boldsymbol{\Phi}(\mathbf{x}'), \tag{3}$$

where $\boldsymbol{\Phi}(\mathbf{x}) \in \mathbb{R}^{2S}$ and

$$\boldsymbol{\Phi}(\mathbf{x})^{(s)} = \begin{cases} \sqrt{a_s}\cos(\boldsymbol{\omega}_s^T\mathbf{x}) & \text{if } 1 \leq s \leq S, \\ \sqrt{a_s}\sin(\boldsymbol{\omega}_s^T\mathbf{x}) & \text{if } S < s \leq 2S. \end{cases}$$

Letting $\boldsymbol{\Lambda} = (\boldsymbol{\Phi}(\mathbf{x}_1), \ldots \boldsymbol{\Phi}(\mathbf{x}_n))$, we have $\mathbf{K_{XX}} \approx \boldsymbol{\Lambda}^T\boldsymbol{\Lambda}$. This allows the use of the efficient matrix determinant and inversion updates (e.g., the Woodbury lemmas (Hager, 1989)) for GP training and prediction using $\mathcal{O}(S^3+Sn)$ runtime and $\mathcal{O}(Sn)$ space. The number of features S is pre-set based on computational concerns, with 100-1000 features typical in applications (Potapczynski et al., 2021; Lázaro-Gredilla et al., 2010; Mutny and Krause, 2018).

Existing methods for Fourier feature approximation differ in their choice of ω_s and a_s. We review two popular approaches used for GP regression below:

- **Random Fourier Features** (RFF; Rahimi and Recht, 2007) approximate the integral in (2) via Monte Carlo, where $\boldsymbol{\omega_s}$ is sampled from $p_{\boldsymbol{\Theta}}(\boldsymbol{\omega})$ and $a_s = 1/S$ so the estimator in (3) is a sample average.
- **Gaussian Quadrature Fourier Features** (Gaussian QFF) approximate the integral in (3) via Gaussian quadrature, where ω_s and a_s are selected from numerical quadrature rules. Mutny and Krause (2018) makes use of Gauss Hermite Fourier feature (GHFF) maps, defining $p_{\boldsymbol{\Theta}}(\omega)$ after a change of variable as the weight function and $h(\omega) = \cos(\boldsymbol{\omega}^T(\mathbf{x} - \mathbf{x}'))$ as the integrand. Such an approach, however, is restricted to GPs with the squared exponential (SE) kernel. Shustin and Avron (2022) make use of Gauss Legendre Fourier feature (GLFF) maps, where after sufficient truncation of the integral, the weight function is constant and $h(\omega) = p_{\boldsymbol{\Theta}}(\boldsymbol{\omega})\cos(\boldsymbol{\omega}^T(\mathbf{x}-\mathbf{x}'))$ is the integrand. Such choices have significant impact on approximation accuracy, as we shall see next.

In what follows, we will exploit the symmetry of Gaussian quadrature rules and the Fourier integrand to eliminate half the nodes when constructing GLFF and GHFF maps. As such, we will derive Gaussian QFF maps using S features from the corresponding $2S$-point quadrature rule.

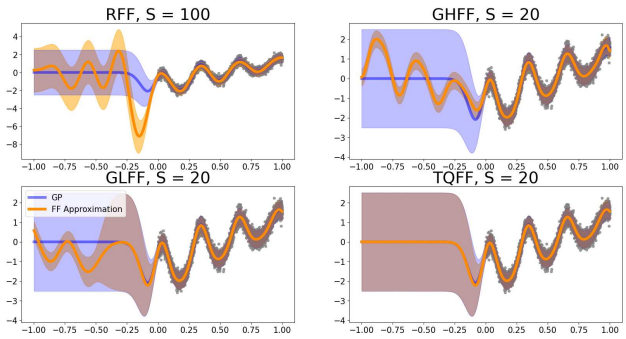

Figure 1: Predictions and 95% predictive intervals from various Fourier feature approximation methods with S features (orange), compared to a fitted full GP (blue). Models share the same hyperparameters optimized from the full GP.

2.4 Drawbacks of Current Fourier Feature Methods

We illustrate the advantages and drawbacks of current Fourier Feature methods via a toy example. Figure 1 shows the fits (using $n = 5000$ training samples) from a full GP with the SE kernel and its various Fourier feature approximations[2]. All methods adopt the same hyperparameters $\boldsymbol{\Theta}$ obtained via maximization of the full GP marginal likelihood. We see clearly that the RFF and GHFF suffer from the aforementioned "variance starvation" in the data sparse region, whereas GLFF performs slightly better. Figure 2 shows the corresponding approximations of $k_{\boldsymbol{\Theta}}(\tau)$ from each method. RFF has trouble representing near zero covariance values between distant points (large τ) due to the slow decay of Monte Carlo error, which results in a severe underestimation of posterior uncertainty in regions far away from the data. Therefore, *covariance saturation* may be a more accurate term for the deteriorating performance of RFF. On the other hand, GHFF and GLFF (the deterministic feature maps) return large errors when estimating covariances for certain values of τ. Such discrepancies explain the poorly calibrated uncertainty quantification properties in Figure 1.

One explanation for the large errors of Gaussian QFF is its reliance on polynomial interpolation to approximate the sinusoidal integrand in Equation 2, which becomes increasingly oscillatory with large τ or low length-scales θ. Figure 3 shows the interpolants and integrands implied by these quadrature feature maps, when approximating $k_{\boldsymbol{\Theta}}(1.75)$ with $S = 15$ features. The GHFF interpolant clearly yields a poor approximation of the integrand, whereas the GLFF interpolant performs better; this is not surprising, as

[2]Experimental details can be found in Appendix 8.

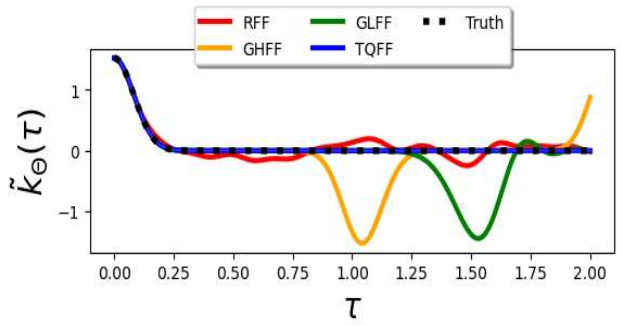

Figure 2: Kernel approximations using various Fourier feature approximation methods, as a function $\tau = x - x'$ for the toy example in Figure 1.

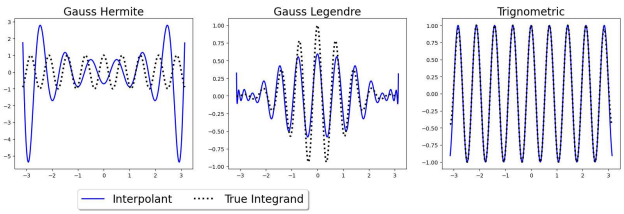

Figure 3: Integrands and interpolants implied by the quadrature rules approximating $k_{\mathbf{\Theta}}(1.75)$ with $S = 15$ features.

Gauss-Legendre quadrature incorporates the rapidly decaying spectral density into the integrand, which damps the oscillatory behavior. These observations suggest that QFF can be greatly improved if one factors in the specific quadrature approximation problem of interest. In doing so, we show next that we can retain the Gaussian QFF's ability to express near-zero covariances (i.e., at small τ) without sacrificing large errors at large τ or at small length-scales.

3 RELATED WORK

Quadrature of highly oscillatory integrals has been studied extensively in the applied mathematics literature (Deaño et al., 2017). Gaussian quadrature is widely known to be sub-optimal for oscillatory quadrature problems, which has motivated alternate approaches including Filon (Huybrechs, 2015), Levin (Huybrechs and Olver, 2009), and steepest descent quadrature (Deaño et al., 2017). While such methods are highly accurate, these quadrature rules cannot be adapted into viable feature maps for kernel approximation. Milovanović et al. (2006, 2008); Da Fies and Vianello (2012) study quadrature rules that are exact for trigonometric polynomials. However, such rules involve solving complex systems of equations that may not have solutions for many weight functions. Furthermore, guaranteeing exactness for all trigonometric polynomials can be wasteful in our setting, and results in inefficient feature approximation maps.

Fourier Feature Approximations for GPs Our work is most related to the GP developments in Mutny and Krause (2018); Shustin and Avron (2022). Mutny and Krause (2018) proposed Gauss Hermite QFF for learning GPs in Bayesian optimization. Recently, Shustin and Avron (2022) made use of Gauss Legendre QFF (along with a rigorous method for choosing sample size dependent hyperparameters) to guarantee spectral equivalence between the full kernel covariance and its low-rank approximation. Potapczynski et al. (2021) further analyzed RFF methods, and found they systematically overfit to data. Significant work has been done on RFF methods for non-GP-related kernel approximation tasks. Avron et al. (2016) analyze error bounds for Quasi Monte Carlo sampled feature maps, while Yu et al. (2016); Munkhoeva et al. (2018) propose sampling from restricted geometries to achieve Monte Carlo variance.

4 TRIGONOMETRIC QUADRATURE FOURIER FEATURES

To address the aforementioned limitations of existing RFF and QFF methods, we propose a new Trigonometric Quadrature Fourier Feature (TQFF) approach. The core idea is to derive a quadrature rule via a cosine interpolant specifically tailored for Fourier transform integrand in Equation (2). We first derive this quadrature rule, then provide its kernel approximation error bounds. We defer all proofs to Appendix 11.

4.1 Kernel Assumptions

We first make the following assumptions on the stationary kernel covariance function $k_{\mathbf{\Theta}}(\cdot, \cdot)$:

Assumption 1. *The stationary kernel covariance function $k_{\mathbf{\Theta}}(\cdot, \cdot)$ satisfies the following:*

(a) The kernel can be written exactly as:

$$k_{\mathbf{\Theta}}(\mathbf{x} - \mathbf{x}') = g(\mathbf{\Theta}) \int p(\boldsymbol{\omega}) \exp(i\boldsymbol{\omega}^T \mathbf{D}(\mathbf{\Theta})(\mathbf{x} - \mathbf{x}'))d\boldsymbol{\omega}$$

for some scalar function $g(\mathbf{\Theta})$, matrix-valued function $\mathbf{D}(\mathbf{\Theta})$, and density $p(\boldsymbol{\omega})$ with no dependency on $\mathbf{\Theta}$.

(b) The density $p(\boldsymbol{\omega})$ in (a) factors over dimensions such that $p(\boldsymbol{\omega}) = \prod_{j=1}^{d} p^{(j)}(\boldsymbol{\omega}^{(j)})$.

The first assumption states that the kernel is the Fourier dual of a spectral density, and we can perform a change-of-variables such that the density does not depend on kernel hyper-parameters. The second assumption is quite common, and can be found in seminal works (Dao et al., 2017; Mutny and Krause, 2018; Avron et al., 2016). Both assumptions are satisfied by common kernel choices, including the SE, 1-d Matérn, and the product-Matérn kernels.

4.2 Trigonometrically Exact Quadrature

For exposition, we begin with the one-dimensional case, which we will later generalize to higher dimensions. From Assumption 1, we can write k_Θ as:

$$k_\Theta(x,x') = g(\Theta)\int_{-\infty}^{\infty} p(\omega)\exp\left(i\omega\left[\frac{x-x'}{\theta}\right]\right)d\omega$$
$$\approx g(\Theta)\int_{-\pi}^{\pi} p_\gamma(\gamma\omega)\cos\left(\omega\gamma\left[\frac{x-x'}{\theta}\right]\right)d\omega, \qquad (4)$$

where θ is its length-scale parameter and $\gamma > 0$ is a pre-set truncation parameter. While Gaussian quadrature aims to achieve exact approximation for polynomial integrands, our trigonometrically exact quadrature rules will instead be exact for integrals of the form of Equation (4) when $\gamma\left[(x-x')/\theta\right]$ is an integer. This leads us to the following definition of a trigonometrically exact rule for our use-case:

Definition 4.1 (Trigonometric Degree of Exactness)**.** *A quadrature rule $Q_S^c(f)$ has trigonometric exactness of degree K with respect to weight function $p_\gamma(\gamma\boldsymbol{\omega})$ if:*

$$Q_S^c(\cos(\boldsymbol{\omega}^T\mathbf{k})) = \int_{[-\pi,\pi]^d} p_\gamma(\gamma\boldsymbol{\omega})\cos(\boldsymbol{\omega}^T\mathbf{k}))d\boldsymbol{\omega}$$

for $\mathbf{k}\in\mathbb{N}^d$ and $||\mathbf{k}||_\infty \le K$.

We next derive a trigonometrically exact rule in one-dimension, by interpolating the integrand $\cos\left(\omega\gamma\left[(x-x')/\theta\right]\right)$ using cosine polynomials and integrating the interpolant against the weight function. A cosine polynomial of degree L, $p_L^c(\omega)$, has the form $p_L^c(\omega) = b_0 + \sum_{l=1}^L b_l\cos(\omega)^l$. We call a cosine polynomial $p_L^c(x)$ *monic* if $b_L = 1$. The unique cosine polynomial $P_{L-1}^c(\omega)$ of degree $L-1$ that interpolates $f(\omega)$ at L distinct nodes $\{\omega_l\}_{l=1}^L, \omega_l\in[0,\pi)$ has the form $P_{L-1}^c(\omega) = \sum_{l=1}^L f(\omega_l)t_l^c(\omega)$, where:

$$t_l^c(\omega) = \prod_{1\le j\le L, j\ne l}\frac{\cos(\omega)-\cos(\omega_j)}{\cos(\omega_l)-\cos(\omega_j)} \qquad (5)$$

Due to the existence of Chebyshev polynomials and uniqueness of the interpolating polynomial, if $f(\omega) = \cos(k\omega)$ and $k\le L-1$, $k\in\mathbb{N}$, then $P_L^c(\omega) = \cos(k\omega)$. As shown in Figure 3, this family of cosine polynomials interpolate the Fourier transform integrand much better than polynomial interpolants of similar degrees.

This family of interpolants leads to an L-point quadrature rule that achieves a trigonometric exactness of degree $2L-1$. We formalize this rule in a proposition.

Proposition 1 (1-d Trigonometric Quadrature)**.** *Adopt the conditions in Assumption 1. Further let $\{q_l^c(\omega)\}_{l=0}^L$ be a sequence of orthogonal monic cosine polynomials with degree l such that:*

$$\int_{-\pi}^{\pi} q_l^c(\omega)q_{l'}^c(\omega)p_\gamma(\gamma\omega)d\omega = 0 \quad \textit{if and only if } l'\ne l.$$

Let $\{\omega_i\}_{i=1}^L$ be the L unique, real-valued zeroes of $q_L^c(\omega)$ in $[0,\pi)$, and define $t_i^c(\omega)$ as in Equation 5. Then, with $a_l = \int_{-\pi}^{\pi} t_l^c(\gamma\omega)p(\gamma\omega)d\omega \ge 0$, the quadrature rule $Q_L^c(f) = \sum_{l=1}^L a_l f(\omega_l)$ has trigonometric exactness of degree $2L-1$.

The kernel approximation derived from the $2L-1$-degree exact trigonometric quadrature rule will be equal to the truncated integral when $\gamma\left[\frac{x-x'}{\theta}\right] \le 2L-1$ and is an integer. We note that our choice to only consider exactness for cosine polynomials increases the efficiency of our quadrature. The rules proposed by Milovanović et al. (2006, 2008) that are exact for general trigonometric polynomials require $2L$ nodes to achieve degree exactness $2L-1$.

Unlike rules that guarantee general trigonometric degree of exactness, the nodes and weights that satisfy the conditions of Proposition 1 can be computed via classical tools from numerical quadrature. In what follows, we use the the popular Golub-Welsh algorithm Golub and Welsch (1969) to find such quadrature nodes and weights (Conte and Boor, 1980). Details on implementation details and computational complexity are provided in supplementary materials.

4.3 Multi-dimensional Extension

Exploiting the assumption (Assumption 1(b)) that the spectral density factors across dimensions, we can use tensor product quadrature to extend our 1-d rule to a trigonometrically exact d-dimensional rule:

Proposition 2 (Multi-dimensional TQFF)**.** *Let $\{(\omega_{i,j}, a_{i,j})\}_{i=1}^L$ denote the L-point trigonometrically exact quadrature rules in dimension j, $j=1,\dots,d$, as defined by Proposition 1. Define new nodes $\omega_{-i,j} = -\omega_{i,j}$ associated with weights $a_{-i,j} = a_{i,j}$ for all $j=1,\dots,p$. Let $\mathcal{S}$ be the largest set of multi-indices $\mathbf{i} = (i_1,\dots,i_d), i_k\in\{-L,\dots,L\}$ such that $\mathcal{S}\subset\prod_{j=1}^d\{-L,\dots,L\}$ but $\mathbf{i}\in\mathcal{S} \implies -\mathbf{i}\notin\mathcal{S}$. Then, with $a_{\mathbf{i}} = \prod_{j=1}^d \frac{1}{2}a_{i_j,j}$ and $\boldsymbol{\omega}_k = (\omega_{\mathbf{i}_1,1},\dots\omega_{\mathbf{i}_d,d})^T$, the*

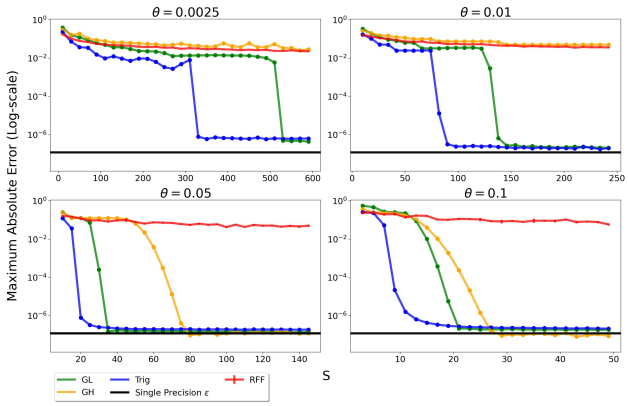

Figure 4: Averaged absolute errors of various Fourier feature maps with S features in approximating the 1-d SE kernel $k_{\mathbf{\Theta}}(\tau)$. This average is taken over a grid of τ values over $[0, 1]$, and θ is the kernel length-scale.

quadrature rule $Q_L(f) = \sum_{\mathbf{i}\in\mathcal{S}} 2a_{\mathbf{i}} f(\boldsymbol{\omega}_{\mathbf{i}})$ *has trigonometric exactness of degree* $2L - 1$.

Maintaining trigonometric exactness of degree $2K - 1$ in d dimensions requires $(2K)^d/2$ total points. Because of this, our method suffers the same curse-of-dimensionality present in other tensor product quadrature methods, and should thus be applied only to applications in lower dimensions or with lower-dimensional structures, e.g., GPs with additive kernels (Duvenaud et al., 2011; Lu et al., 2022).

4.4 Error Bound for Kernel Approximation

We now provide uniform bounds on the approximation error from TQFF:

Proposition 3 (TQFF Error Bound)**.** *Let* $\mathbf{\Phi}(\mathbf{x}), \mathbf{x} \in [0, 1]^d$ *be the feature map derived from the quadrature rule that has trigonometric exactness of degree* $2L - 1$ *in each dimension. Define* $M = \lceil \frac{\gamma}{\min_j \boldsymbol{\theta}_j} \rceil$. *Let* $C_d(\mathbf{\Theta}) = g(\mathbf{\Theta}) d 2^{d-1}$. *Then, for any* $\mathbf{x}, \mathbf{x}' \in [0, 1]^d$*:*

$$|k_{\mathbf{\Theta}}(\mathbf{x}, \mathbf{x}') - \mathbf{\Phi}(\mathbf{x})^T \mathbf{\Phi}(\mathbf{x}')| \leq 2C_d(\mathbf{\Theta}) \int_{\pi}^{\infty} p_\gamma(\gamma\omega) + C_d(\mathbf{\Theta}) \frac{\left[\pi + 4 + 2\ln\left(\frac{2}{\pi}(4L-1)\right)\right] \max\{M, 2L-1\}!}{2(2L)^{\max\{1, 2L-M\}}(M-1)!}.$$

The first term in this bound is the truncation error, and the last term captures quadrature error for the truncated integral. We note that this truncation error rapidly decays with γ. For example, with the SE kernel and its associated Gaussian spectral density, this truncation error can approximately be bounded by floating point single precision at $\gamma = 1.15$.

Another interesting observation is that the TQFF error bound depends only linearly on the minimum inverse length-scale $1/\min_j\{\boldsymbol{\theta}_j\}$, rather than quadratically as in the GHFF bound in Mutny and Krause (2018); Dao et al. (2017). Although the TQFF approximation error decays much faster than RFF, this decay is asymptotically slightly slower than the exponential decrease obtained via Gaussian quadrature. However, we shall see that, empirically, this shortcoming is insignificant in the single precision setting prevalent in machine learning.

Explicit error bound comparison with GLFF (Shustin and Avron, 2022) is difficult due to their unique measure of convergence. We instead compare these errors empirically. Figure 4 shows the average absolute error[3] of these methods when approximating the SE kernel $k_{\mathbf{\Theta}}(\tau)$ for $\tau \in [0, 1]$. Here, γ is set at 1.15 for both TQFF and GLFF so that their error approximately converges to floating point single precision (single precision is used here as it is the default for popular GP implementations (Gardner et al., 2018; Matthews et al., 2017), is computationally efficient, and produces similar accuracy to double precision (Maddox et al., 2022)). We see that, for each length-scale setting, the average error of the proposed TQFF converges quickest over all methods, with the next best method (GLFF) requiring at least $\approx 50\%$ more features to achieve errors near single precision ϵ. TQFF yields significantly smaller average errors throughout the pre-convergence period. GLFF, on the other hand, yields higher average error than RFF and TQFF for low S. RFF can be seen to converge slowly, and GHFF struggles with lower length-scales. Discrepancy between floating point precision and TQFF/GLFF convergence can be attributed to numerical errors in computing quadrature rules (Laurie, 2001). GHFF does not suffer as heavily from these errors due to the implementation of specialized algorithms for computing Gauss-Hermite rules (Townsend et al., 2016).

5 NUMERICAL EXPERIMENTS

We empirically evaluate TQFF against existing Fourier feature approximations for GP regression. Such comparisons are focused primarily on Fourier feature approximation methods, as they possess properties uniquely desirable in a wide range of applications. We do, however, include the Sparse Gaussian Process Regression (SGPR; (Titsias, 2009)) as a standard baseline. To compare against the popularly-implemented GHFF approximation, all methods in this section will use an anisoptric SE kernel. All models are trained

[3]We provide empirical analysis of maximum absolute error plots and error distributions in Appendix 9.

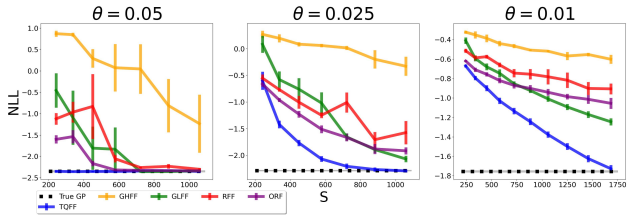

Figure 5: Test NLL for the compared methods for the 2-d synthetic function sampled from a GP prior. Error bars indicate ± 1 standard error over 5 random seeds.

using the Adam optimizer in PyTorch (Kingma and Ba, 2017; Paszke et al., 2019). For GLFF and TQFF, γ is fixed at 1.15 to bound truncation error at single precision. Discussion of the effect of γ on kernel approximation error can be found in Appendix 10.

5.1 2-d Synthetic Functions from GP

We explore here the effectiveness of TQFF for 2-d synthetic functions. We first generate training data of sample size $n = 20,000$, using functions simulated from a 2-d GP prior with isotropic SE kernels, with different length-scales $\theta = .05, .025$ and $.01$. Predictions are made on test sets of $4,000$ samples. We then compare the learned Fourier feature approximations to a full GP with hyperparameters set as the ground truth. We evaluate the test negative-log-likelihood (NLL) of the compared methods for various feature sample sizes S. This procedure is then replicated 5 times.

Figure 5 shows the test NLL of each method for different length-scales θ. We see that TQFF outperforms competing methods for all θ and S by converging fastest to the full GP performance. For $\theta = .05$, TQFF requires noticeably less features than GLFF and RFF to achieve comparable performance to the full GP. GLFF also performs relatively well for large S, but is often outperformed by RFF for small S.

5.2 Approximation of Posterior Uncertainty

We now examine the performance of TQFF for approximating GP posterior uncertainty, which is crucial for applications such as Bayesian optimization (Chen et al., 2023) and surrogate modeling (Li et al., 2023; Ji et al., 2022). We adopt the solar irradiance reconstruction experiment in Lean et al. (1995); Gal and Turner (2015); Hensman et al. (2018), where we removed 5 segments from a time series dataset (representing solar irradiance) and examined the predictive distribution in these hold-out segments (see Figure 6). The same methods as before are compared here, with the quadrature Fourier feature approximation methods using $S = 70$ features and RFF using $S = 300$ features. The SGPR baseline is fit with $m = 140$ inducing points.

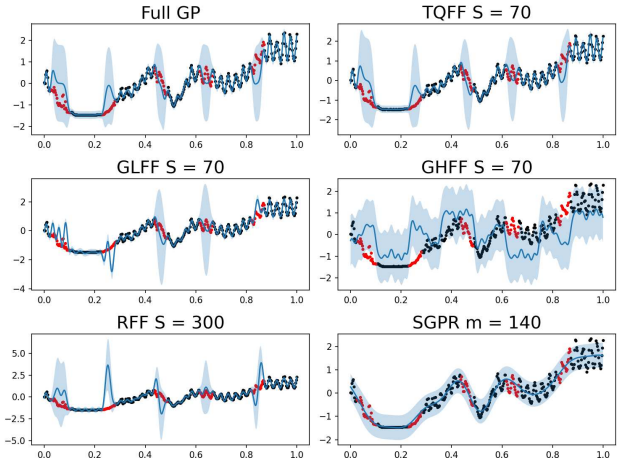

Figure 6: Posterior predictive means and 95% predictive intervals for the compared methods in the solar irradiance application. The training data is in marked in black, while the hold-out data is in red.

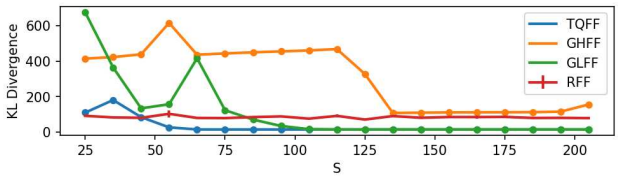

Figure 7: KL-divergences of the predictive distribution of the full GP from the compared approximation methods, at hold-out data points in the solar irradiance application. Error bars for RFF indicate ±1 standard error over 5 random seeds.

Figure 6 shows the predictions of the compared methods, with its 95% predictive intervals. We see the posterior predictive distribution from our TQFF is visually indistinguishable from the desired full GP posterior. GLFF and RFF perform well in regions with training data, but suffers from variance starvation in regions with hold-out data, resulting in notable undercoverage. GHFF performs poorly here, due to the aforementioned difficulty of Gauss-Hermite quadrature at low length-scales. SGPR appears to oversmooth the true function in regions of high oscillations, which is undesirable.

We can quantify this performance by examining the KL-divergence of the full GP predictive distribution on the hold-out points from the predictive distribution

generated from the Fourier Feature approximations. Figure 7 shows this KL divergence as a function of the number of features S, where RFF results are averaged over 5 random seeds to account for sampling variation. TQFF yields lower KL-divergence from the full GP with far fewer features than the other methods. The KL-divergence of RFF and GHFF appear to converge slowly in S, while GLFF requires many features ($S > 100$) to achieve near-zero KL-divergence.

5.3 Regression Benchmarks

We compare TQFF on several commonly-used low-dimensional GP regression benchmarks. We examine the time-series dataset of Google daily stock prices (Ton et al., 2018; Shustin and Avron, 2022), a household electricity consumption dataset (Hebrail and Berard, 2012), a UK Apartment Housing price dataset (Hensman et al., 2013), and the commonly-used Schaffer function benchmark (Surjanovic and Bingham, 2010). Each dataset is randomly split into 80% training and 20% testing. We examine the performance of the Fourier approximation methods up to $S = 1058$ features. SGPR is also included with the number of inducing points set to $m = 2S$ in order to match the computational compelity of the Fourier feature methods. Dataset details can be found in Appendix 13.

Figure 8 shows the RMSE and NLL over 5 random seeds as a function of the number of features S. We see that, for small S, RFF may outperform the compared quadrature approximation methods. Once a moderate S is achieved however, the proposed TQFF yields the lowest RMSE and NLL among the considered Fourier feature methods across all datasets. We also see that, for $d = 2$, the feature efficiency of TQFF relative to Gaussian QFF increases for larger S, as the number of total nodes scales quadratically with the size of the 1-d rules. This is consistent with findings in Shustin and Avron (2022), who noted that GLFF often requires significantly more than a thousand features before outperforming RFF on real data with low length-scales. SGPR outperforms TQFF in terms of RMSE on the very low-length-scale HouseElectric dataset. However, TQFF still significantly outperforms SPGR in terms of NLL here. This difference in uncertainty quantification performance is likely due to the fact that TQFF targets the full GP model via a numerical approximation of the full covariance kernel. In contrast, SGPR leverages a variational approximation to a sparse GP model, which is known in the literature to sometimes over-estimate uncertainty (Bauer et al., 2016; Jankowiak et al., 2020).

It may seem peculiar that RFF outperforms QFF methods for smaller S, when Figure 4 shows that QFF often has lower average kernel approximation error. This may be explained by the error distribution discussed in Appendix 9, which shows that QFF methods can have larger error tails for small S. Regardless, the lower average error for the proposed TQFF allows for improved performance over its Gaussian QFF counterparts for smaller S, and its fast convergence enables superior performance over RFF for larger S.

6 DISCUSSION

We proposed a new trigonometric quadrature Fourier feature (TQFF) method for scalable GP modeling. The key idea behind TQFF is the use of a novel trigonometric quadrature rule, specifically tailored for the desired Fourier transform in Fourier feature approximation. In doing so, this addresses the known limitation of variance starvation for existing Fourier feature methods in GP approximation. We provide approximation error bounds for TQFF, and then demonstrate the improved performance of TQFF over competing methods in a suite of numerical experiments and applications. In particular, we show the TQFF enjoys accurate approximations (with well-calibrated uncertainties) for GPs over a broad range of length-scales using fewer features.

Our promising results suggest a new class of Fourier feature maps that can be derived from custom interpolants and integrands for the desired Fourier transform integrand. An interesting future direction would be the use of Bayesian quadrature with a kernel designed for the trigonometric integrand, to achieve better accuracy in higher dimensions. Applying recent methods for highly-oscillatory quadrature (Deaño et al., 2017) may also be fruitful for improving accuracy.

A drawback of the TQFF (along with general QFF approaches) is the curse-of-dimensionality. A potential solution might be to extend the TQFF to higher dimensions via sparse grid quadrature, as explored in Dao et al. (2017). This extension would benefit from the improved efficiency of TQFF over Gaussian QFF, and we will explore this as future work. The extension of TQFF for problems with inherent low-dimensional structure is also of great interest, particularly via additive kernels (Duvenaud et al., 2011; Lu et al., 2022).

Acknowledgements

The authors gratefully thank the reviewers for their judicious suggestions and acknowledge funding from NSF CSSI 2004571, NSF DMS 2210729, NSF DMS 2316012 and DE-SC0024477.

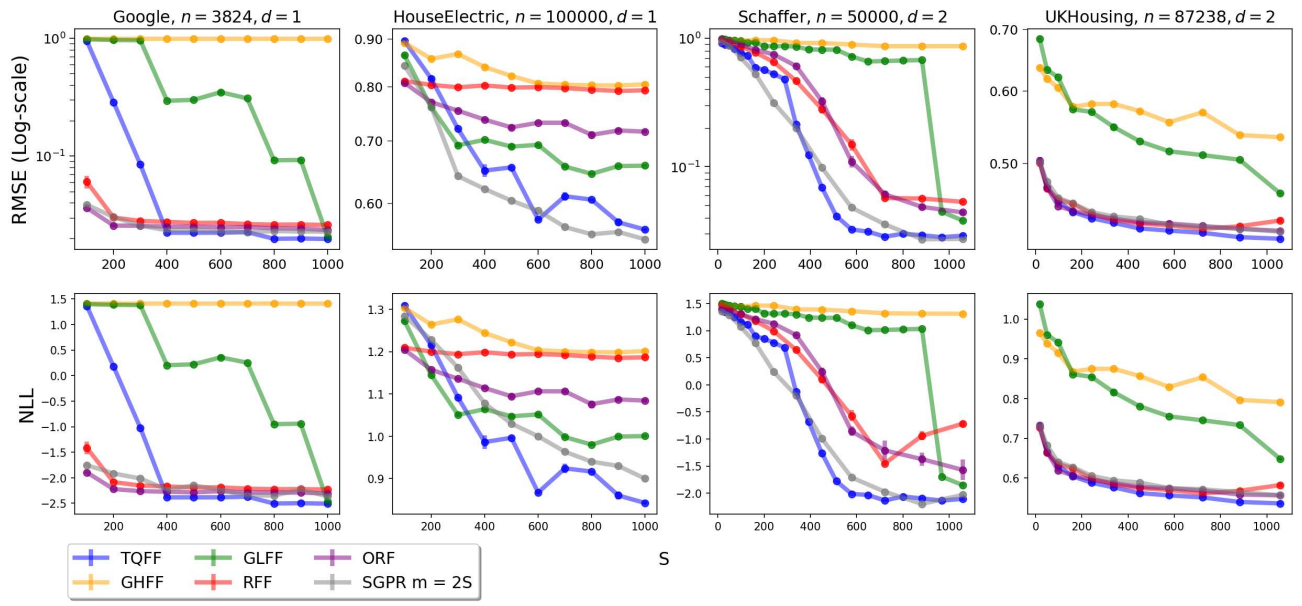

Figure 8: Average test RMSE and NLL on different regression benchmark datasets. Error bars indicate ±1 standard error over 5 random seeds. Training sample size n and data dimension d are outlined in plot title.

References

Angelis, E., Wenk, P., Schölkopf, B., Bauer, S., and Krause, A. (2020). SLEIPNIR: deterministic and provably accurate feature expansion for gaussian process regression with derivatives. *CoRR*, abs/2003.02658.

Avron, H., Sindhwani, V., Yang, J., and Mahoney, M. W. (2016). Quasi-monte carlo feature maps for shift-invariant kernels. *J. Mach. Learn. Res.*, 17(1):4096–4133.

Bauer, M., van der Wilk, M., and Rasmussen, C. E. (2016). Understanding probabilistic sparse gaussian process approximations. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Cao, J., Kang, M., Jimenez, F., Sang, H., Schaefer, F. T., and Katzfuss, M. (2023). Variational sparse inverse cholesky approximation for latent Gaussian processes via double kullback-leibler minimization. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 3559–3576. PMLR.

Chen, Z., Mak, S., and Wu, C. F. J. (2023). A hierarchical expected improvement method for bayesian optimization. *Journal of the American Statistical Association*, pages 1–14.

Conte, S. D. and Boor, C. W. D. (1980). *Elementary Numerical Analysis: An Algorithmic Approach*. McGraw-Hill Higher Education, 3rd edition.

Cutajar, K., Bonilla, E. V., Michiardi, P., and Filippone, M. (2017). Random feature expansions for deep Gaussian processes. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 884–893. PMLR.

Da Fies, G. and Vianello, M. (2012). Trigonometric gaussian quadrature on subintervals of the period. *Electron. Trans. Numer. Anal*, 39:102–112.

Dai, Z., Low, B. K. H., and Jaillet, P. (2020). Federated bayesian optimization via thompson sampling. 33:9687–9699.

Dai, Z., Low, B. K. H., and Jaillet, P. (2021). Differentially private federated bayesian optimization with distributed exploration. 34:9125–9139.

Dao, T., De Sa, C. M., and Ré, C. (2017). Gaussian quadrature for kernel features. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Deaño, A., Huybrechs, D., and Iserles, A. (2017). *Computing highly oscillatory integrals*. SIAM.

Deisenroth, M. and Ng, J. W. (2015). Distributed gaussian processes. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1481–1490, Lille, France. PMLR.

Delvos, F.-J. (1993). Hermite interpolation with trigonometric polynomials. *BIT Numerical Mathematics*.

Deng, Y., Zhou, X., Kim, B., Tewari, A., Gupta, A., and Shroff, N. (2022). Weighted gaussian process bandits for non-stationary environments. 151:6909–6932.

Dubey, A. (2021). No-regret algorithms for private gaussian process bandit optimization. In Banerjee, A. and Fukumizu, K., editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2062–2070. PMLR.

Duvenaud, D. K., Nickisch, H., and Rasmussen, C. (2011). Additive gaussian processes. *Advances in neural information processing systems*, 24.

Gal, Y. and Turner, R. (2015). Improving the gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 655–664, Lille, France. PMLR.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. (2018). Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Golub, G. H. and Welsch, J. H. (1969). Calculation of gauss quadrature rules. *Mathematics of computation*, 23(106):221–230.

Hager, W. W. (1989). Updating the inverse of a matrix. *SIAM review*, 31(2):221–239.

Hebrail, G. and Berard, A. (2012). Individual household electric power consumption. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C58K54.

Hensman, J., Durrande, N., and Solin, A. (2018). Variational fourier features for gaussian processes. *Journal of Machine Learning Research*, 18(151):1–52.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, page 282–290, Arlington, Virginia, USA. AUAI Press.

Huybrechs, D. (2015). *Filon Quadrature*, pages 513–516. Springer Berlin Heidelberg, Berlin, Heidelberg.

Huybrechs, D. and Olver, S. (2009). Highly oscillatory quadrature. *Highly oscillatory problems*, 366:25–50.

Jankowiak, M., Pleiss, G., and Gardner, J. (2020). Parametric Gaussian process regressors. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4702–4712. PMLR.

Ji, Y., Yuchi, H. S., Soeder, D., Paquet, J.-F., Bass, S. A., Joseph, V. R., Wu, C., and Mak, S. (2022). Multi-stage multi-fidelity Gaussian process modeling, with application to heavy-ion collisions. *arXiv preprint arXiv:2209.13748*.

Katzfuss, M. and Guinness, J. (2021). A General Framework for Vecchia Approximations of Gaussian Processes. *Statistical Science*, 36(1):124 – 141.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Kunitsa, V. (1970). The remainder term of a trigonometric interpolation expression for equally spaced nodes in spectral form. *Cybernetics*, 6:605–615.

Laurie, D. P. (2001). Computation of gauss-type quadrature formulas. *Journal of Computational and Applied Mathematics*, 127(1-2):201–217.

Lázaro-Gredilla, M., Quiñnero-Candela, J., Rasmussen, C. E., and Figueiras-Vidal, A. R. (2010). Sparse spectrum gaussian process regression. *Journal of Machine Learning Research*, 11(63):1865–1881.

Lean, J., Beer, J., and Bradley, R. (1995). Reconstruction of solar irradiance since 1610: Implications for climate change. *Geophysical Research Letters*, 22(23):3195–3198.

Li, K., Mak, S., Paquet, J.-F., and Bass, S. A. (2023). Additive multi-index Gaussian process modeling, with application to multi-physics surrogate modeling of the quark-gluon plasma. *arXiv preprint arXiv:2306.07299*.

Lin, J. A., Antorán, J., Padhy, S., Janz, D., Hernández-Lobato, J. M., and Terenin, A. (2023). Sampling from gaussian process posteriors using stochastic gradient descent.

Lu, X., Boukouvalas, A., and Hensman, J. (2022). Additive gaussian processes revisited. In *International Conference on Machine Learning*, pages 14358–14383. PMLR.

Maddox, W. J., Potapcynski, A., and Wilson, A. G. (2022). Low-precision arithmetic for fast gaussian processes. In *Uncertainty in Artificial Intelligence*, pages 1306–1316. PMLR.

Matthews, A. G. d. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. (2017). GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6.

Milovanović, G. V., Cvetković, A. S., and Stanić, M. P. (2006). Trigonometric orthogonal systems and quadrature formulae with maximal trigonometric degree of exactness. In *International Conference on Numerical Methods and Applications*, pages 402–409. Springer.

Milovanović, G. V., Cvetković, A. S., and Stanić, M. P. (2008). Trigonometric orthogonal systems and quadrature formulae. *Computers & Mathematics with Applications*, 56(11):2915–2931.

Munkhoeva, M., Kapushev, Y., Burnaev, E., and Oseledets, I. (2018). Quadrature-based features for kernel approximation. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Mutny, M. and Krause, A. (2018). Efficient High Dimensional Bayesian Optimization with Additivity and Quadrature Fourier Features. 31.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Potapczynski, A., Wu, L., Biderman, D., Pleiss, G., and Cunningham, J. P. (2021). Bias-free scalable gaussian processes via randomized truncations. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8609–8619. PMLR.

Qing, J., Dhaene, T., and Couckuyt, I. (2022). Spectral representation of robustness measures for optimization under input uncertainty. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18096–18121. PMLR.

Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Ray Chowdhury, S. and Gopalan, A. (2019). Bayesian optimization under heavy-tailed payoffs. 32.

Shustin, P. F. and Avron, H. (2022). Gauss-legendre features for gaussian process regression. *Journal of Machine Learning Research*, 23(92):1–47.

Snelson, E. and Ghahramani, Z. (2005). Sparse gaussian processes using pseudo-inputs. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press.

Surjanovic, S. and Bingham, D. (2010). Virtual library of simulation experiments: Test functions and datasets.

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In van Dyk, D. and Welling, M., editors, *Proceedings of the Twelth 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. PMLR.

Ton, J.-F., Flaxman, S., Sejdinovic, D., and Bhatt, S. (2018). Spatial mapping with gaussian processes and nonstationary fourier features. *Spatial Statistics*, 28:59–78. One world, one health.

Townsend, A., Trogdon, T., and Olver, S. (2016). Fast computation of gauss quadrature nodes and weights on the whole real line. *IMA Journal of Numerical Analysis*, 36(1):337–358.

V.V Ivanov, V. Z. (1966). Certain New Results in Approximation Theory . *Vychislitel'naya Matematika*.

Wang, K., Pleiss, G., Gardner, J., Tyree, S., Weinberger, K. Q., and Wilson, A. G. (2019). Exact gaussian processes on a million data points. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Wang, Z., Gehring, C., Kohli, P., and Jegelka, S. (2018). Batched large-scale bayesian optimization in high-dimensional spaces. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 745–754. PMLR.

Warren, H., Oliveira, R., and Ramos, F. (2022). Generalized bayesian quadrature with spectral kernels. In Cussens, J. and Zhang, K., editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 2085–2095. PMLR.

Wilson, J., Borovitskiy, V., Terenin, A., Mostowsky, P., and Deisenroth, M. (2020). Efficiently sampling functions from Gaussian process posteriors. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10292–10302. PMLR.

Wilson, J. T., Borovitskiy, V., Terenin, A., Mostowsky, P., and Deisenroth, M. P. (2021). Pathwise conditioning of gaussian processes. *Journal of Machine Learning Research*, 22(105):1–47.

Wu, L., Pleiss, G., and Cunningham, J. P. (2022). Variational nearest neighbor Gaussian process. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 24114–24130. PMLR.

Yu, F. X. X., Suresh, A. T., Choromanski, K. M., Holtmann-Rice, D. N., and Kumar, S. (2016). Orthogonal random features. *Advances in neural information processing systems*, 29.

Zhang, M. M., Gundersen, G. W., and Engelhardt, B. E. (2023). Bayesian non-linear latent variable modeling via random fourier features.

Zhang, M. M. and Williamson, S. A. (2019). Embarrassingly parallel inference for gaussian processes. *Journal of Machine Learning Research*, 20(169):1–26.

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** We will include implementation in jupyter notebooks within the supplementary material.
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. **Yes**
   (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). **No**
   (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**

   We will include these details in the supplementary material.
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.**Not Applicable**
   (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 partici-

pants 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 Material

7 Tensor Product Quadrature Rule

The 1-dimensional quadrature rules can be extended to higher dimensions via tensor products. If we assume that a multi-dimensional integral factors across dimensions and we apply a L point quadrature rule in each dimension, we can write:

$$\int_a^b p(\boldsymbol{\omega})h(\boldsymbol{\omega})d\omega = \prod_{j=1}^{d}\int_a^b p^{(j)}(\boldsymbol{\omega}^{(j)})f^{(j)}(\boldsymbol{\omega}^{(j)})d\omega^{(j)}$$

$$\approx \prod_{j=1}^{d}\sum_{l=1}^{L} a_{l,j}h(\omega_{l,j}) = \sum_{\mathbf{l}\in\prod_{j=1}^{d}\{1\dots L\}} a_{\mathbf{l}}h(\boldsymbol{\omega}_{\mathbf{l}})$$

where $a_{l,j}$, $\omega_{l,j}$ are the lth quadrature node and weight in dimension j respectively. $a_{\mathbf{l}} = \prod_{j=1}^{d} a_{\mathbf{l}^{(j)},j}$ and $\boldsymbol{\omega}_{\mathbf{l}} = (\omega_{\mathbf{l}^{(1)},1}, \dots \omega_{\mathbf{l}^{(d)},d})^T$. Clearly, the number of quadrature nodes grows exponentially with dimensions, which limits tensor product quadrature to problems in low dimensions or with low dimensional structure.

8 Toy Example Details

In the example in Section 2.4, we draw $n = 5000$ samples from the set-up

$$y_i = \exp(-x_i^2)\exp(\sin(10(x_i - .5))^2) + 3x_i + \epsilon, \epsilon \sim \mathcal{N}(0, .1^2)$$

The plots shown in the paper are normalized so that the output has unit standard deviation and zero mean. We draw $x \sim U(0, 1)$ and make predictions for 1000 x^* sampled such that $x^* \sim U(-1, 1)$. GLFF and TQFF are given truncation parameter $\gamma = 1.15$ to bound truncation error at floating point precision.

9 Maximum Approximation Error and Error Distribution

The average kernel approximation error does not tell the full story. Figure 9 shows the maximum approximation error of the methods for the SE kernel $k_{\boldsymbol{\Theta}}(\tau)$. The maximum is taken over a $n = 100$ grid of τ defined on the unit intereval. For small L we see that RFF performs better. However, only the QFF methods are able to quickly converge to near single precision ϵ error. The difference between convergence and single precision ϵ can be attributed to numerical errors for calculating the quadrature rule and kernels.

We further examine the error distribution for the methods. Figure 10 shows the distribution of absolute errors for the methods when approximating the SE kernel $k_{\boldsymbol{\Theta}}(\tau)$ for length-scale $\theta = .01$ using $S = 25$ features. The approximations are made on a grid of τ over the unit interval. We see that the QFF methods have long tails, while RFF does not suffer from the very large errors. However, we see that the errors of TQFF are highly concentrated around zero, with skinnier error tails relative to the Gaussian QFF methods. TQFF error is also more concentrated around zero than RFF which is consistent with the lower average error we observe.

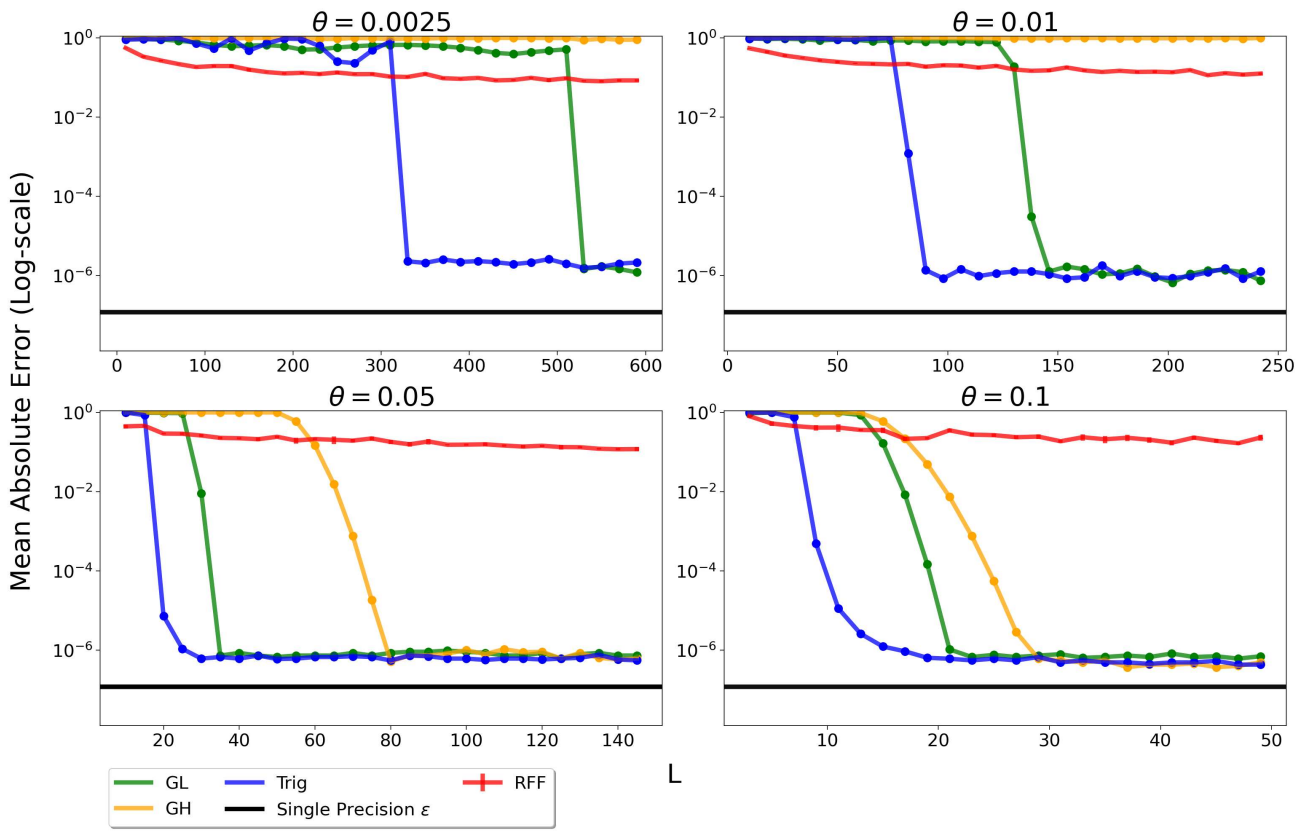

Figure 9: Maximum absolute kernel approximation error for SE 1d $k_{\Theta}(\tau)$.Maximum taken over $n = 100$ grid of τ defined on unit interval.

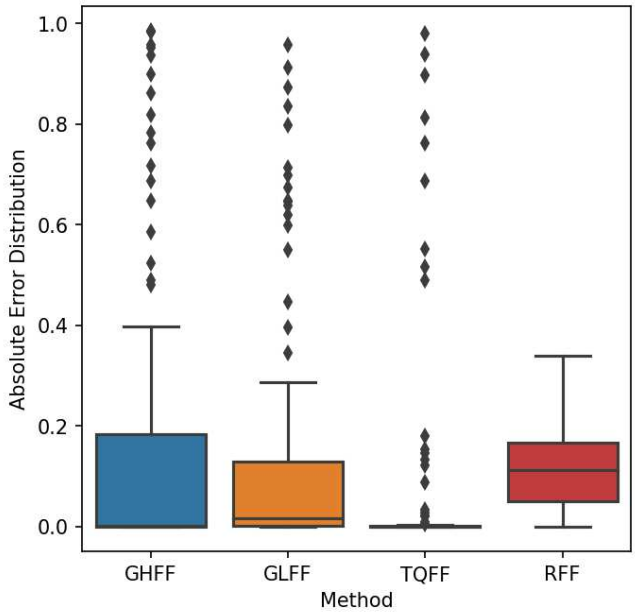

Figure 10: Distribution of absolute errors when approximating an SE kernel $k_{\Theta}(\tau)$ for length-scale $\theta = .01$ using $S = 25$ features. τ is defined over a $n = 100$ grid on the unit interval

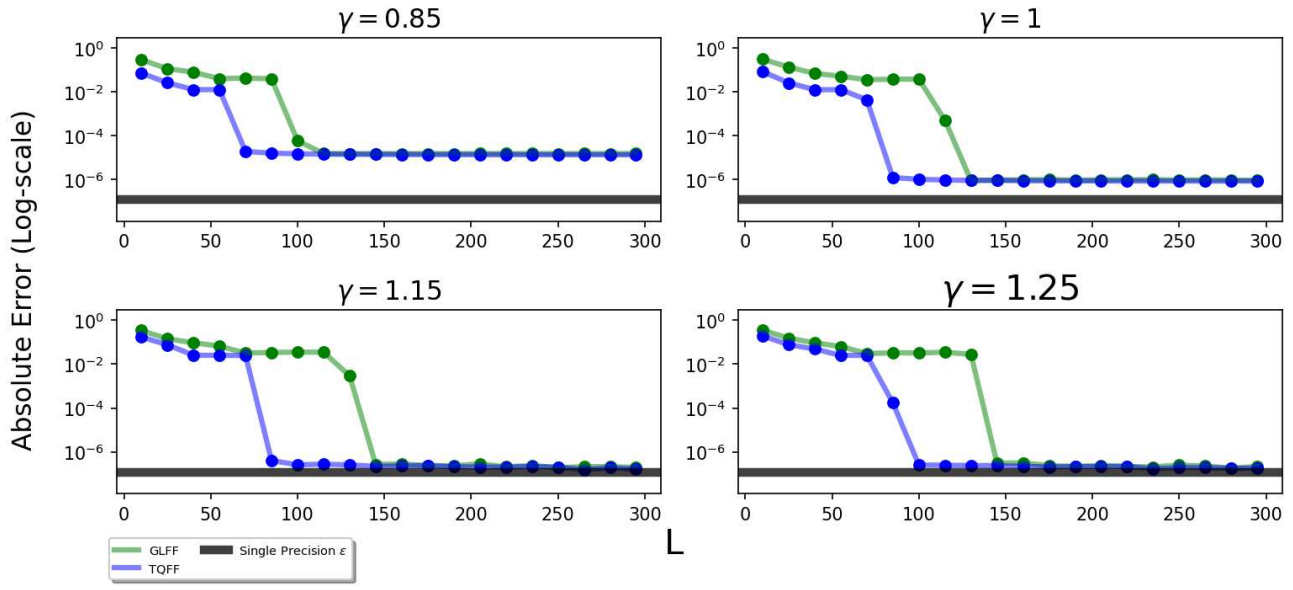

Figure 11: Approximation error as a function of quadrature nodes L for $k_{\mathbf{\Theta}}(1)$ with $\theta = .01$ for various settings of truncation parameter γ.

10 Effect of γ on approximation error

We examine the effect of different values of the truncation parameter γ on the kernel covariance approximation accuracy of TQFF and GLFF. Figure 11 shows the mean absolute approximation error of TQFF and GLFF for the kernel covariance $k_{\mathbf{\Theta}}(\tau)$. The absolute error is averaged over τ on a $n = 100$ grid on the unit interval given various γ. TQFF achieves smaller approximation error across values γ using significantly fewer features. In addition, as γ decreases the approximation error for both methods converges more quickly to the truncation error. This behavior is expected as decreasing γ dampens frequency of the oscillatory integrand. However we also see that lower γ results in convergence to higher errors due to truncation. The fact that TQFF still performs better across γ is expected as dampening the oscillatory behavior benefits all interpolation strategies.

One could also implement a truncated version of RFF and GHFF to find an optimal balance between quadrature and truncation error for each Fourier Feature approximation. However, this is beyond the scope of this paper and we defer this question to future work. For fair comparison, in the remainder of the paper we set $\gamma = 1.15$ so that the truncation error is upper bounded by approximately single precision machine ϵ.

11 Proofs

Note that all the proofs here are in extremely similar flavor to the proofs for standard Gaussian quadrature.A good reference is Conte and Boor (1980).

11.1 Uniqueness and Exactness of Interpolating Cosine Polynomial

Let $P^c_{L-1}(\omega)$ be the degree $L-1$ interpolating cosine polynomial of $f(\omega)$ through points $\{\omega_i\}_{i=1}^L$. Define polynomial function $T_{L-1}(z)$ of degree $L-1$ such that $T_{L-1}(cos(\omega)) = P^c_{L-1}(\omega)$ which is possible as we restrict $\omega \in [0, \pi)$. Suppose that there exists another polyonmial $H_{L-1}(z)$ such that $T_{L-1}(z_i) = H_{L-1}(z_i)$ for all $z_i = cos(\omega_i), i = 1 \dots n$. Then the polynomial $T_{L-1}(z_i) - H_{L-1}(z_i)$ has L zeros at $cos(\omega_i)$. This is impossible as $T_{L-1}(z) - H_{L-1}(z)$ are degree $L-1$. As $cos(\omega)$ is injective on $[0, \pi)$ this contradiction shows that there does not exist another degree $L-1$ polynomial function of cosine that interpolates $f(\omega)$ through these L points.

Uniqueness also implies that $P^c_{L-1}(\omega) = f(\omega)$ when $f(\omega)$ is a cosine polynomial of degree $L-1$.

11.2 Proof of Proposition 1

First we note that for any choice of quadrature nodes $\{\omega_l\}_{l=1}^{L}$, the an L point quadrature rule will have trignometric degree of exactness $L-1$. To prove exactness note that is $f(\omega)$ is a $L-1$ degree cosine polynomial we can write

$$Q_L(f) = \sum_{i=1}^{L} a_i f(\omega_i) = \sum_{i=1}^{L} \int_{-\pi}^{\pi} t_i^c(\omega) f(\omega_i) p_\gamma(\gamma\omega) d\omega = \int_{-\pi}^{\pi} f(\omega) p_\gamma(\gamma\omega) d\omega$$

As all functions of form $cos(k\omega), k \in \mathbb{N}$ can be written as cosine polynomials the statement follows. Now we derive the parts of Proposition 1

11.2.1 Existence of zeros of $q_L^c(\omega)$

First note that by construction $q_L^c(\omega)$ is orthogonal to all cosine polynomials with degree $< L$, i.e

$$\int_{-\pi}^{\pi} q_L^c(\omega) p_l^c(\omega) p_\gamma(\gamma\omega) d\omega = 0$$

For all cosine polynomials $p_l^(\omega)$ with degree $l < L$. Now we can prove that $q_L^c(\omega)$ has L zeros in $[0, \pi)$. First by construction we have that

$$\int_{-\pi}^{\pi} q_L^c(\omega) p_\gamma(\gamma\omega) d\omega = 0$$

Therefore $q_L^c(\omega)$ changes sign for at least one ω in $[0, \pi)$. Because $q_L^c(\omega)$ is a cosine polynomial, it has at most L zeroes in $[0, \pi)$. Assume that $q_L^c(\omega)$ has $1 \leq m < L$ distinct zeroes on $[0, \pi)$ located at $\omega_1 \dots \omega_m$ of odd multiplicity (aka when $q_L^c(\omega)$ changes sign). Define the degree M cosine polynomial:

$$Z_L(\omega) = \prod_{i=1}^{M} (cos(\omega) - cos(\omega_i))$$

By construction $Z_L(\omega) q_L^c(\omega)$ does not change sign on the integration interval and therefore $\int_{-\pi}^{\pi} Z_L(\omega) q_L^c(\omega) p(\gamma\omega) \neq 0$. However, this is a contradiction as by assumption $q_L^c(\omega)$ is orthogonal to all cosine polynomials with degree $< L$. Therefore $q_L^c(\omega)$ has at least L zeroes on $[0, \pi)$. But because it is a cosine polynomial it has at most L zeroes there, so we are done.

11.2.2 Exactness

Let $f(\omega)$ be a cosine polynomial of degree $2L-1$. We can use standard polynomial division to obtain

$$f(\omega) = q_L^c(\omega) q(\omega) + r(\omega)$$

Where $q(\omega), r(\omega)$ are cosine polynomials of degree $\leq L-1$. Therefore we write the integral

$$\int_{-\pi}^{\pi} f(\omega) p_\gamma(\gamma\omega) d\omega = \int_{-\pi}^{\pi} q_L^c(\omega) q(\omega) p_\gamma(\gamma\omega) d\omega + \int_{-\pi}^{\pi} r(\omega) p_\gamma(\gamma\omega) d\omega$$
$$= \int_{-\pi}^{\pi} r(\omega) p_\gamma(\gamma\omega) d\omega$$

Define our L-point trignometrically exact quadrature rule with nodes located at the zeroes of $q_L^c(\gamma\omega)$ in $[0, \pi)$. We have that $Q_L(r(\omega)) = \int_{-\pi}^{\pi} r(\omega) p(\gamma\omega) d\omega$ immediately by the exactness of L-point quadrature. Because

$q_L^c(\omega_i)q(\omega_i) = 0$ by the choice of quadrature nodes:

$$\int_{-\pi}^{\pi} f(\omega)p_\gamma(\gamma\omega)d\omega = \int_{-\pi}^{\pi} r(\omega)p_\gamma(\gamma\omega)d\omega = \sum_{i=1}^{L} a_i r(\omega_i) = \sum_{i=1}^{L} a_i(q_L^c(\omega_i)q(\omega_i) + r(\omega_i))$$

$$= \sum_{i=1}^{L} a_i f(\omega_i) = Q_L(f)$$

Which shows exactness.

11.2.3 Positivity of a_i

Recall our definition of $a_i = \int_{-\pi}^{\pi} t_i^c p(\gamma\omega)d\omega$. If we employ our quadrature rule of exactness degree $2L-1$ we have that

$$\int_{-\pi}^{\pi} (t_i^c)^2 p_\gamma(\gamma\omega)d\omega = Q_L((t_i^c)^2) = \sum_{k=1}^{L} a_k(t_k^c(x_k))^2 = a_i$$

because $(t_i^c(\omega))^2$ has degree $2L-2$ so that clearly $a_i > 0$.

11.3 Proof of Proposition 2

Let $Q_L(f)$ be defined as in Proposition 2 in d dimensions. Suppose $f(\boldsymbol{\omega}) = cos(\boldsymbol{\omega}^T\mathbf{k})$ for $\mathbf{k} \in \mathbb{N}^d$ and $||\mathbf{k}||_\infty \leq L-1$. $\boldsymbol{\omega}^{(j)}$ refers to the j-th element of $\boldsymbol{\omega} \in \mathbb{R}^d$. This is distinct from the $\boldsymbol{\omega}_\mathbf{i} \in \mathbb{R}^d$ where the elements of $\boldsymbol{\omega}_\mathbf{i}$ are constructed according to the multi-index $\mathbf{i}$ as stated in the proposition.

$$Q_L(f) = \sum_{\mathbf{i}\in\mathcal{S}} 2a_\mathbf{i} f(\boldsymbol{\omega}_i) = \sum_{\mathbf{i}\in\prod_{j=1}^d\{-L,\dots L\}} a_\mathbf{i} \exp(i\boldsymbol{\omega}_\mathbf{i}^T\mathbf{k})$$

$$= \prod_{j=1}^{d} \sum_{i=-L, i\neq 0}^{L} \frac{1}{2} a_{i,j} \exp(i\omega_{i,j}\mathbf{k}^{(j)}) = \prod_{j=1}^{d} \int_{-\pi}^{\pi} p_j(\gamma\boldsymbol{\omega}^{(j)}) \exp(i\boldsymbol{\omega}^{(j)}\mathbf{k}^{(j)})d\boldsymbol{\omega}^{(j)}$$

$$= \int_{[-\pi,\pi]^d} p_\gamma(\gamma\boldsymbol{\omega}) \exp(i\boldsymbol{\omega}^T\mathbf{k})d\boldsymbol{\omega} = \int_{[-\pi,\pi]^d} p_\gamma(\gamma\boldsymbol{\omega}) \cos(\boldsymbol{\omega}^T\mathbf{k})d\boldsymbol{\omega}$$

Giving us the desired notion of trignometric exactness. The equality in the second line follows from the enforced symmetry of our nodes in each dimension. We write

$$\sum_{i=-L, i\neq 0}^{L} \frac{1}{2} a_{i,j} \exp(i\omega_{i,j}k_j) = \sum_{i=-L, i\neq 0}^{L} \frac{1}{2} a_{i,j}(\cos(\omega_{i,j}k_j) + i\sin(\omega_{i,j}k_j)) = \sum_{i=-L, i\neq 0}^{L} \frac{1}{2} a_{i,j} \cos(\omega_{i,j}k_j)$$

$$= \sum_{i=1}^{L} a_{i,j} \cos(\omega_{i,j}k_j) = \int_{-\pi}^{\pi} p_\gamma(\gamma\boldsymbol{\omega}_j) cos(\boldsymbol{\omega}_j k_j)d\omega_j = \int_{-\pi}^{\pi} p_\gamma(\gamma\boldsymbol{\omega}^{(j)}) \exp(i\boldsymbol{\omega}^{(j)}\mathbf{k}^{(j)})d\boldsymbol{\omega}^{(j)}$$

11.4 Proof of Proposition 3

11.4.1 Necessary Results

We first state a necessary theorem regarding the error of trignometric interpolation from (Kunitsa, 1970; V.V Ivanov, 1966)

Proposition 4 (Ivanov 1966)**.** *Let $f(\omega)$ be a r- times differentiable function. Then for any trigonometric polynomial $p_K(\omega)$ of degree K that interpolates $f(\omega)$ at $K+1$ dinstinct points in $[-\pi,\pi]$ we have that*

$$|f(x) - p_K(x)| \leq \left[\frac{\pi}{2} + 2 + ln\left(\frac{2}{\pi}(2K+1)\right)\right] \times \frac{\sup_{|z(\omega)|=1} |f_K^{(r)}(z(\omega))|}{(K+1)^r}$$

Where $z(\omega) = e^{i\omega}$ and $p_K^{(r)}(z(\omega))$ is the r-th (complex valued) derivative of $f_L(\cdot)$ with respect to $z(\omega)$.

And a small modification of a relevant theorem for generalizing 1-dimensional quadrature errors to tensor product quadrature from (Mutny and Krause, 2018)

Proposition 5 (Mutny 2018)**.** *Let $\boldsymbol{\omega} \in \mathbb{R}^d$ and $\mathbf{k} \in \mathbb{R}^d$. Under the assumptions, suppose that the error of a one dimensional quadrature rule approximation to the integral $\int p_j(\boldsymbol{\omega}_j\gamma)\cos(\boldsymbol{\omega}_j\mathbf{k}_j)d\boldsymbol{\omega}_j$ can be bounded by ϵ. Then the tensor product quadrature error for $\int p(\boldsymbol{\omega}\gamma)\cos(\boldsymbol{\omega}^T\mathbf{k})d\boldsymbol{\omega}$ scales as $\epsilon d 2^{d-1}$.*

We can write the original integral as:

$$\int p(\boldsymbol{\omega}\gamma)\cos(\boldsymbol{\omega}^T\mathbf{k})d\boldsymbol{\omega} = \int p(\boldsymbol{\omega}\gamma)\exp(i\boldsymbol{\omega}^T\mathbf{k})d\boldsymbol{\omega} = \prod_{j=1}^{d}\int p_j(\boldsymbol{\omega}_j\gamma)\cos(\boldsymbol{\omega}_j\mathbf{k}_j)$$

If we can upper bound the error for approximating each integral in the product of the last inequality of ϵ, we can apply lemma 7 from Mutny and Krause (2018) and the error of approximating the original integral is $d2^{d-1}\epsilon$.

11.4.2 Proof

First we examine the one dimensional case. We want to bound the error of the L point quadrature of trigonometric degree of exactness $K = 2L-1$. Error bounds on one dimension can then be extended to the multiple dimensions by observing that the one dimensional quadratures involved in the multidimensional extension are exactly equal in value to the standard L point quadrature and therefore we can apply theorem 5.

First we write the form of our feature map/kernel approximation:

$$\begin{aligned}
k_\theta(x,x') &= \int_{-\infty}^{\infty} p_{\boldsymbol{\Theta}}(\omega)\cos(\omega(x-x'))d\omega \approx g(\boldsymbol{\Theta})\int_{-\pi}^{\pi} p_\gamma(\gamma\omega)\cos\left(\omega\gamma\left[\frac{x-x'}{\theta}\right]\right)d\omega \\
&\approx g(\boldsymbol{\Theta})Q_L\left(\cos\left(\omega\gamma\left[\frac{x-x'}{\theta}\right]\right)\right)d\omega) = g(\boldsymbol{\Theta})\sum_{i=1}^{L} a_i(\cos(\omega_i\frac{x}{\theta})\cos(\omega_i\frac{x'}{\theta}) + \sin(\omega_i\frac{x}{\theta})\sin(\omega_i\frac{x'}{\theta})) \\
&= \boldsymbol{\Phi}(\mathbf{x})^T\boldsymbol{\Phi}(\mathbf{x}')
\end{aligned}$$

Where

$$\boldsymbol{\Phi}(x)_i = \begin{cases} \sqrt{a_i\gamma g(\boldsymbol{\Theta})}cos(\omega_i\gamma\frac{x}{\theta}) & \text{if } i \le L \\ \sqrt{a_{i-L}\gamma g(\boldsymbol{\Theta})}cos(\omega_{i-L}\gamma\frac{x}{\theta}) & \text{if } L < i \le 2L \end{cases}$$

Accuracy of the feature map approximation is clearly exactly the accuracy of the quadrature. Defining $p_\gamma(\omega) = \gamma p(\omega)$, we write

$$k_\theta(x,x') = g(\boldsymbol{\Theta})\int_{-\pi}^{\pi} p_\gamma(\gamma\omega)\cos\left(\omega\gamma\left[\frac{x-x'}{\theta}\right]\right)d\omega + 2g(\boldsymbol{\Theta})\int_{\pi}^{\infty} p_\gamma(\gamma\omega)\cos\left(\omega\gamma\left[\frac{x-x'}{\theta}\right]\right)d\omega$$

From now on, define $\alpha = \gamma\left[\frac{x-x'}{\theta}\right]$. Given our truncation parameter γ, we will apply quadrature to the first integral. We can use trigonometric Hermite polynomials (Delvos (1993), Propositions 4.1 and 4.2) to define a degree K trigonometric polynomial of form

$$p_K^t(\omega) = c_0 + \sum_{l=1}^{K} c_l\cos(l\omega) + d_l\sin(l\omega)$$

Such that $p_K^t(\omega_i) = cos(\alpha\omega_i)$, $i = 1\dots L$ where $\{\omega_i\}_{i=1}^{L}$ are the zeros of the monic orthogonal trignometric polynomial of degree L. $p_K^t(\omega)$ is of degree $K = 2L-1$ and the terms involving $sin(\omega)$ integrate to zero over the symmetric domain and weighting function. Therefore the integral and can be exactly integrated by our quadrature rule:

$$\begin{aligned}
&\int_{-\pi}^{\pi} p_\gamma(\gamma\omega)\cos(\alpha\omega)\,d\omega = \int_{-\pi}^{\pi} p_\gamma(\gamma\omega)p_K^t(\omega)d\omega + \int_{-\pi}^{\pi} p_\gamma(\gamma\omega)(\cos(\alpha\omega) - p_K^t(\omega))d\omega \\
&= Q_L(cos(\alpha\omega)) + \int_{-\pi}^{\pi} p_\gamma(\gamma\omega)(\cos(\alpha\omega) - p_K^t(\omega))d\omega
\end{aligned}$$

The total error becomes

$$
\begin{aligned}
&|k_\theta(x,x') - \mathbf{\Phi}(x)^T\mathbf{\Phi}(x')| = |\gamma g(\mathbf{\Theta})\left(\int_{-\infty}^{\infty} p(\gamma\omega)cos(\alpha\omega)d\omega - Q_L(cos(\alpha\omega))\right)| \\
&= g(\mathbf{\Theta})|\int_{-\pi}^{\pi} p_\gamma(\gamma\omega)(\cos(\alpha\omega) - p_K^t(\omega))d\omega + 2\int_{\pi}^{\infty} p_\gamma(\gamma\omega)cos(\alpha\omega)d\omega| \\
&\leq g(\mathbf{\Theta})\left(\int_{-\pi}^{\pi} p_\gamma(\gamma\omega)|cos(\alpha\omega) - p_K^t(\omega)|d\omega + 2\int_{\pi}^{\infty} p_\gamma(\gamma\omega)d\omega\right)
\end{aligned}
$$

We need to bound $|cos(\alpha\omega) - p_K^t(\omega)|$. Using theorem 4, letting $M = \lceil \frac{\gamma}{\theta} \rceil \geq \alpha$ and setting $r = \max\{2L - M, 1\}$ we have that $p_K^t(\omega)$ also satisfies

$$
|f(\omega) - p_{2L-1}^c(\omega)| \leq \left[\frac{\pi}{2} + 2 + \ln\left(\frac{2}{\pi}(4L-1)\right)\right] \times \frac{\sup_{|z(\omega)|=1}|f_L^{(r)}(z(\omega))|}{(2L)^r}
$$

Define $z(\omega) = \exp(i\omega)$ and notice that

$$
cos(\alpha\omega) = \frac{1}{2}\left(\exp(i\omega)^\alpha + \exp(-i\omega)^\alpha\right) = \frac{1}{2}(z(\omega)^\alpha + z(\omega)^{-\alpha}) := f(z(\omega))
$$

Note that because we restrict the supremum to ω such that $|z(\omega)| = 1$, repeated differentiation of $z(\omega)^\alpha, z(\omega)^{-\alpha}$ wrt $z(\omega)$ gives us

$$
\sup_{|z(\omega)|=1} |f_L^{(r)}(z(\omega))| \leq \frac{(\lceil\alpha\rceil + r - 1)!}{(\lceil\alpha\rceil - 1)!} \leq \frac{(M + r - 1)!}{(M-1)!}
$$

.

Assuming that $x, x' \in [0,1]$, $\alpha \leq \lceil \frac{\gamma}{\theta} \rceil = M$. Therefore our final bound can be written, definint $H = \max\{2L - 1, M\}$

$$
|k_\theta(x,x') - \mathbf{\Phi}(x)^T\mathbf{\Phi}(x')| \leq g(\mathbf{\Theta})\left(\left[\frac{\pi}{2} + 2 + \ln\left(\frac{2}{\pi}(4L-1)\right)\right] \times \frac{\max\{M, 2L-1\}!}{(2L)^{\max\{1,2L-M\}}(M-1)!} + 2\int_{\pi}^{\infty} p_\gamma(\gamma\omega)\right)
$$

The extension to the multi-dimensional case involves the tensor product of one-dimensional rules that produce identical error to the 1-dimensional quadrature produced here. Therefore we can apply theorem 5 and the result follows.

12 Computation of Nodes and Weights in the Golub Welsh Algorithm

Recall that we need to compute a monic cosine polynomial of degree L $q_L^c(\omega)$ that is orthogonal to all cosine polynomials of degree less than L. We can use Three term recurrence relation and the Golub-Welsh algorithm for this task (Conte and Boor, 1980). The monic orthonormal cosine polynomials $\{q_l(\omega)\}$ associated with the weight function $p(\omega)$ on $[a,b]$ satisfy the relation:

$$
\begin{aligned}
q_1(\omega) &= (cos(\omega) - B_0)q(\omega) \\
q_{k+1}(\omega) &= (cos(\omega) - B_k)q_k(\omega) - A_k(q_{k-1}(\omega))
\end{aligned}
$$

where $A_k = \frac{||q_k||^2}{||q_{k-1}||^2}$, $k \geq 1$ and $B_k = \frac{\langle cos(\omega)q_k, q_k\rangle}{||q_k^c||^2}, k \geq 0$. Where the inner product is $\langle f, g\rangle = \int_a^b f(\omega)g(\omega)p(\omega)d\omega$. All inner products involve integrals of powers of $cos(\omega)$ against the weight function $p(\omega)$ and can be calculated analytically using a software such as Mathematica or Maple. Analytical solutions exist for kernels such as the RBF and matern.

Because cosine polynomials can be written as polynomials of functions defined on the interval $[-1,1]$, we can apply the standard Golub-Welsh algorithm using the eigenvalues/eigen vectors of the tri-diagonal matrix formed

from B_k, A_k to obtain nodes/weights that satisfy the conditions of Proposition 1. Please see Conte and Boor (1980) or any numerical analysis text for more details. The only difference is we have to take the inverse cosine transformation of the eigenvalues of the matrix to get our nodes.

Implementation of Golub-Welsh and TTR requires $\mathcal{O}(L^3)$ complexity to compute L nodes. This computational cost is not too burdensome as the quadrature rules only have to be computed once independent of any dataset. We note that this approach works well for obtaining quadrature rules and fourier features up to $L \approx 1000$ which is the upper bound for most applications. Numerical error begins to accumulate at this point and computation is burdensome. To efficiently produce more features one can easily apply asymptotic type methods from standard Gaussian quadrature to scale to hundreds of thousands of features (Townsend et al., 2016).

13 Benchmark Data Sources

For the google data we obtained the log daily high stock price from `https://finance.yahoo.com/quote/GOOG?p=GOOG`. We took data from 9/11/2004 - 9/13/2023. The Household electricity data-set was taken from the frist 125,000 observations from the dataset stored in `https://archive.ics.uci.edu/dataset/235/individual+household+electric+power+consumption`. UKHousing data was obtained from the 2018 price paid dataset of sale prices filtered for flats (apartments) located at `https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads`. We extracted the prices from May-December. The Schaffer function is a widely used benchmark function. Specification can be found at `https://www.sfu.ca/~ssurjano/schaffer2.html`. We evaluate the function on the hyercube $[-3, 3]^2$.

References

Angelis, E., Wenk, P., Schölkopf, B., Bauer, S., and Krause, A. (2020). SLEIPNIR: deterministic and provably accurate feature expansion for gaussian process regression with derivatives. *CoRR*, abs/2003.02658.

Avron, H., Sindhwani, V., Yang, J., and Mahoney, M. W. (2016). Quasi-monte carlo feature maps for shift-invariant kernels. *J. Mach. Learn. Res.*, 17(1):4096–4133.

Bauer, M., van der Wilk, M., and Rasmussen, C. E. (2016). Understanding probabilistic sparse gaussian process approximations. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Cao, J., Kang, M., Jimenez, F., Sang, H., Schaefer, F. T., and Katzfuss, M. (2023). Variational sparse inverse cholesky approximation for latent Gaussian processes via double kullback-leibler minimization. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 3559–3576. PMLR.

Chen, Z., Mak, S., and Wu, C. F. J. (2023). A hierarchical expected improvement method for bayesian optimization. *Journal of the American Statistical Association*, pages 1–14.

Conte, S. D. and Boor, C. W. D. (1980). *Elementary Numerical Analysis: An Algorithmic Approach*. McGraw-Hill Higher Education, 3rd edition.

Cutajar, K., Bonilla, E. V., Michiardi, P., and Filippone, M. (2017). Random feature expansions for deep Gaussian processes. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 884–893. PMLR.

Da Fies, G. and Vianello, M. (2012). Trigonometric gaussian quadrature on subintervals of the period. *Electron. Trans. Numer. Anal*, 39:102–112.

Dai, Z., Low, B. K. H., and Jaillet, P. (2020). Federated bayesian optimization via thompson sampling. 33:9687–9699.

Dai, Z., Low, B. K. H., and Jaillet, P. (2021). Differentially private federated bayesian optimization with distributed exploration. 34:9125–9139.

Dao, T., De Sa, C. M., and Ré, C. (2017). Gaussian quadrature for kernel features. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Deaño, A., Huybrechs, D., and Iserles, A. (2017). *Computing highly oscillatory integrals*. SIAM.

Deisenroth, M. and Ng, J. W. (2015). Distributed gaussian processes. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1481–1490, Lille, France. PMLR.

Delvos, F.-J. (1993). Hermite interpolation with trigonometric polynomials. *BIT Numerical Mathematics*.

Deng, Y., Zhou, X., Kim, B., Tewari, A., Gupta, A., and Shroff, N. (2022). Weighted gaussian process bandits for non-stationary environments. 151:6909–6932.

Dubey, A. (2021). No-regret algorithms for private gaussian process bandit optimization. In Banerjee, A. and Fukumizu, K., editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2062–2070. PMLR.

Duvenaud, D. K., Nickisch, H., and Rasmussen, C. (2011). Additive gaussian processes. *Advances in neural information processing systems*, 24.

Gal, Y. and Turner, R. (2015). Improving the gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 655–664, Lille, France. PMLR.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. (2018). Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Golub, G. H. and Welsch, J. H. (1969). Calculation of gauss quadrature rules. *Mathematics of computation*, 23(106):221–230.

Hager, W. W. (1989). Updating the inverse of a matrix. *SIAM review*, 31(2):221–239.

Hebrail, G. and Berard, A. (2012). Individual household electric power consumption. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C58K54.

Hensman, J., Durrande, N., and Solin, A. (2018). Variational fourier features for gaussian processes. *Journal of Machine Learning Research*, 18(151):1–52.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, page 282–290, Arlington, Virginia, USA. AUAI Press.

Huybrechs, D. (2015). *Filon Quadrature*, pages 513–516. Springer Berlin Heidelberg, Berlin, Heidelberg.

Huybrechs, D. and Olver, S. (2009). Highly oscillatory quadrature. *Highly oscillatory problems*, 366:25–50.

Jankowiak, M., Pleiss, G., and Gardner, J. (2020). Parametric Gaussian process regressors. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4702–4712. PMLR.

Ji, Y., Yuchi, H. S., Soeder, D., Paquet, J.-F., Bass, S. A., Joseph, V. R., Wu, C., and Mak, S. (2022). Multi-stage multi-fidelity Gaussian process modeling, with application to heavy-ion collisions. *arXiv preprint arXiv:2209.13748*.

Katzfuss, M. and Guinness, J. (2021). A General Framework for Vecchia Approximations of Gaussian Processes. *Statistical Science*, 36(1):124 – 141.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Kunitsa, V. (1970). The remainder term of a trigonometric interpolation expression for equally spaced nodes in spectral form. *Cybernetics*, 6:605–615.

Laurie, D. P. (2001). Computation of gauss-type quadrature formulas. *Journal of Computational and Applied Mathematics*, 127(1-2):201–217.

Lázaro-Gredilla, M., Quiñnero-Candela, J., Rasmussen, C. E., and Figueiras-Vidal, A. R. (2010). Sparse spectrum gaussian process regression. *Journal of Machine Learning Research*, 11(63):1865–1881.

Lean, J., Beer, J., and Bradley, R. (1995). Reconstruction of solar irradiance since 1610: Implications for climate change. *Geophysical Research Letters*, 22(23):3195–3198.

Li, K., Mak, S., Paquet, J.-F., and Bass, S. A. (2023). Additive multi-index Gaussian process modeling, with application to multi-physics surrogate modeling of the quark-gluon plasma. *arXiv preprint arXiv:2306.07299*.

Lin, J. A., Antorán, J., Padhy, S., Janz, D., Hernández-Lobato, J. M., and Terenin, A. (2023). Sampling from gaussian process posteriors using stochastic gradient descent.

Lu, X., Boukouvalas, A., and Hensman, J. (2022). Additive gaussian processes revisited. In *International Conference on Machine Learning*, pages 14358–14383. PMLR.

Maddox, W. J., Potapcynski, A., and Wilson, A. G. (2022). Low-precision arithmetic for fast gaussian processes. In *Uncertainty in Artificial Intelligence*, pages 1306–1316. PMLR.

Matthews, A. G. d. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. (2017). GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6.

Milovanović, G. V., Cvetković, A. S., and Stanić, M. P. (2006). Trigonometric orthogonal systems and quadrature formulae with maximal trigonometric degree of exactness. In *International Conference on Numerical Methods and Applications*, pages 402–409. Springer.

Milovanović, G. V., Cvetković, A. S., and Stanić, M. P. (2008). Trigonometric orthogonal systems and quadrature formulae. *Computers & Mathematics with Applications*, 56(11):2915–2931.

Munkhoeva, M., Kapushev, Y., Burnaev, E., and Oseledets, I. (2018). Quadrature-based features for kernel approximation. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Mutny, M. and Krause, A. (2018). Efficient High Dimensional Bayesian Optimization with Additivity and Quadrature Fourier Features. 31.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Potapczynski, A., Wu, L., Biderman, D., Pleiss, G., and Cunningham, J. P. (2021). Bias-free scalable gaussian processes via randomized truncations. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8609–8619. PMLR.

Qing, J., Dhaene, T., and Couckuyt, I. (2022). Spectral representation of robustness measures for optimization under input uncertainty. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18096–18121. PMLR.

Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Ray Chowdhury, S. and Gopalan, A. (2019). Bayesian optimization under heavy-tailed payoffs. 32.

Shustin, P. F. and Avron, H. (2022). Gauss-legendre features for gaussian process regression. *Journal of Machine Learning Research*, 23(92):1–47.

Snelson, E. and Ghahramani, Z. (2005). Sparse gaussian processes using pseudo-inputs. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press.

Surjanovic, S. and Bingham, D. (2010). Virtual library of simulation experiments: Test functions and datasets.

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In van Dyk, D. and Welling, M., editors, *Proceedings of the Twelth 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. PMLR.

Ton, J.-F., Flaxman, S., Sejdinovic, D., and Bhatt, S. (2018). Spatial mapping with gaussian processes and nonstationary fourier features. *Spatial Statistics*, 28:59–78. One world, one health.

Townsend, A., Trogdon, T., and Olver, S. (2016). Fast computation of gauss quadrature nodes and weights on the whole real line. *IMA Journal of Numerical Analysis*, 36(1):337–358.

V.V Ivanov, V. Z. (1966). Certain New Results in Approximation Theory . *Vychislitel'naya Matematika*.

Wang, K., Pleiss, G., Gardner, J., Tyree, S., Weinberger, K. Q., and Wilson, A. G. (2019). Exact gaussian processes on a million data points. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Wang, Z., Gehring, C., Kohli, P., and Jegelka, S. (2018). Batched large-scale bayesian optimization in high-dimensional spaces. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 745–754. PMLR.

Warren, H., Oliveira, R., and Ramos, F. (2022). Generalized bayesian quadrature with spectral kernels. In Cussens, J. and Zhang, K., editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 2085–2095. PMLR.

Wilson, J., Borovitskiy, V., Terenin, A., Mostowsky, P., and Deisenroth, M. (2020). Efficiently sampling functions from Gaussian process posteriors. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10292–10302. PMLR.

Wilson, J. T., Borovitskiy, V., Terenin, A., Mostowsky, P., and Deisenroth, M. P. (2021). Pathwise conditioning of gaussian processes. *Journal of Machine Learning Research*, 22(105):1–47.

Wu, L., Pleiss, G., and Cunningham, J. P. (2022). Variational nearest neighbor Gaussian process. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 24114–24130. PMLR.

Yu, F. X. X., Suresh, A. T., Choromanski, K. M., Holtmann-Rice, D. N., and Kumar, S. (2016). Orthogonal random features. *Advances in neural information processing systems*, 29.

Zhang, M. M., Gundersen, G. W., and Engelhardt, B. E. (2023). Bayesian non-linear latent variable modeling via random fourier features.

Zhang, M. M. and Williamson, S. A. (2019). Embarrassingly parallel inference for gaussian processes. *Journal of Machine Learning Research*, 20(169):1–26.