# Scalable Gaussian Processes with Grid-Structured Eigenfunctions (GP-GRIEF)

Trefor W. Evans and Prasanth B. Nair

University of Toronto

ICML

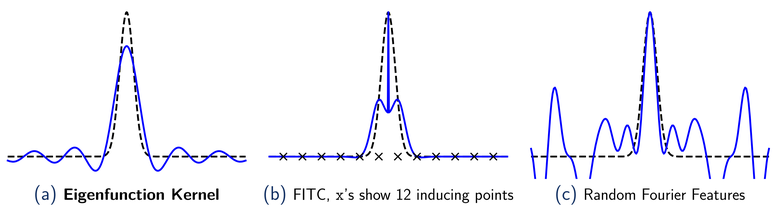
## Overview

We introduce a kernel approximation strategy that enables computation of the Gaussian process log marginal likelihood and all hyperparameter derivatives in $\mathcal{O}(p)$ time. Our GRIEF kernel consists of $p$ eigenfunctions found using a Nyström approximation from a dense Cartesian product grid of inducing points. By exploiting algebraic properties of Kronecker and Khatri-Rao tensor products, computational complexity of the training procedure can be practically *independent* of the number of inducing points. This allows us to use arbitrarily many inducing points to achieve a globally accurate kernel approximation, even in high-dimensional problems. The fast likelihood evaluation enables type-I or II Bayesian inference on large-scale datasets. We benchmark our algorithms on real-world problems with up to two-million training points and $10^{33}$ inducing points.
Code available at `https://github.com/treforevans/gp_grief`

## Eigenfunction Kernel

We approximate an exact kernel as a finite sum of eigenfunctions using a Nyström approximation from an inducing point set [1]. This representation is attractive since

- eigenfunctions give the most compact representation among orthogonal functions;
- the eigenfunctions live in a reproducing kernel Hilbert space;
- the approximate eigenfunctions converge in the limit of large $n$ [2]; and
- the approximate eigenfunctions converge with many inducing points (large $m$), which we will consider. This result is shown in Theorem 1 below.



(a) **Eigenfunction Kernel** (b) FITC, x's show 12 inducing points (c) Random Fourier Features

Figure: Comparison of kernel approximations using $p = 12$ basis functions. Exact kernel shown in black.

We approximate an "exact" kernel $k$ using $p$ eigenfunctions to give

$$\begin{aligned} &\text{the kernel} && \widetilde{k}(\mathbf{x},\mathbf{z}) = \textstyle\sum_{i=1}^{p} (\lambda_i^{-\frac{1}{2}}\mathbf{K}_{\mathbf{x},\mathrm{U}}\mathbf{q}_i)(\lambda_i^{-\frac{1}{2}}\mathbf{K}_{\mathbf{z},\mathrm{U}}\mathbf{q}_i), \text{ and} \\ &\text{the covariance matrix} && \widetilde{\mathbf{K}}_{\mathrm{X,X}} = \mathbf{K}_{\mathrm{X,U}}\mathbf{Q}\mathbf{S}_p^T\mathbf{\Lambda}_p^{-1}\mathbf{S}_p\mathbf{Q}^T\mathbf{K}_{\mathrm{U,X}} \end{aligned}$$

where X is the set of training points, U is the set of inducing points, $\mathbf{Q}, \mathbf{\Lambda} \in \mathbb{R}^{m\times m}$ are unitary and diagonal matrices, respectively, formed from the eigen-decomposition of $\mathbf{K}_{\mathrm{U,U}} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, and $\mathbf{S}_p \in \mathbb{R}^{p\times m}$ is a sparse selection matrix. The following result shows why we want to use lots of inducing points (large $m$) since the *approximate* eigenfunctions become *exact*.

### Theorem 1: Eigenfunction Convergence

If the $i$th eigenvalue of $k$ is simple and non-zero and U $\supset$ X, a Nyström approximation of the $i$th kernel eigenfunction converges in the limit of large $m$,

$$\mathbf{q}_i^{(n)} = \lim_{m\to\infty} \sqrt{\frac{m}{n}} \frac{1}{\lambda_i^{(m)}} \mathbf{K}_{\mathrm{X,U}}\mathbf{q}_i^{(m)},$$

where $\lambda_i^{(m)} \in \mathbb{R}$ and $\mathbf{q}_i^{(m)} \in \mathbb{R}^m$ are the $i$th largest eigenvalue and corresponding eigenvector of $\mathbf{K}_{\mathrm{U,U}}$, respectively. $\mathbf{q}_i^{(n)}$ is the kernel eigenfunction corresponding to the $i$th largest eigenvalue, evaluated on the set X.

## Gridded Inducing Points

To use a large $m$, we place inducing points on a Cartesian grid to fill out the input space while allowing many more inducing points than training points ($m \gg n$). The grid contains $\overline{m}=\sqrt[d]{m}\approx\mathcal{O}(10)$ points along each dimension. The covariance matrix then inherits the Kronecker product ($\otimes$) structure $\mathbf{K}_{\mathrm{U,U}}=\bigotimes_{i=1}^{d}\mathbf{K}_{\mathrm{U,U}}^{(i)}$, enabling efficient Kronecker matrix algebra to be exploited [3]. For instance,

$\mathbf{K}_{\mathrm{U,U}}=\bigotimes_{i=1}^{d}\mathbf{K}_{\mathrm{U,U}}^{(i)}$ storage $\rightarrow \mathcal{O}(d\overline{m}^2)$
$\mathbf{K}_{\mathrm{U,U}}=\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ factoring $\rightarrow \mathcal{O}(d\overline{m}^3)$
$\mathbf{Q} = \bigotimes_{i=1}^{d}\mathbf{Q}^{(i)}$ MVM $\rightarrow \mathcal{O}(d\overline{m}^{d+1})$

## Exponential Scaling

In low-dimensions, exploiting gridded inducing point structure can be greatly advantageous, however, we can immediately see in the block to the left that complexity of MVMs with $\widetilde{\mathbf{K}}_{\mathrm{X,X}}$ increases *exponentially* in $d$! This poor scaling poses a serious impediment to the successful application of the proposed approach, or SKI [3], to high-dimensional datasets. We next discuss how to overcome this computational bottleneck.

### Theorem 2: Linear Scaling

The product of a row-partitioned Khatri-Rao matrix $\mathbf{K}_{\mathrm{X,U}} = *_{i=1}^{d}\mathbf{K}_{\mathrm{X,U}}^{(i)}$, a Kronecker product matrix $\mathbf{Q} = \bigotimes_{i=1}^{d}\mathbf{Q}^{(i)}$, and a column-partitioned Khatri-Rao matrix $\mathbf{S}_p^T = *_{i=1}^{d}(\mathbf{S}_p^{(i)})^T$ can be computed as follows

$$\underbrace{\mathbf{K}_{\mathrm{X,U}}}_{\mathbb{R}^{n\times\overline{m}^d}}\underbrace{\mathbf{Q}}_{\mathbb{R}^{\overline{m}^d\times\overline{m}^d}}\underbrace{\mathbf{S}_p^T}_{\mathbb{R}^{\overline{m}^d\times p}} = \bigodot_{i=1}^{d}\underbrace{\mathbf{K}_{\mathrm{X,U}}^{(i)}}_{\mathbb{R}^{n\times\overline{m}}}\underbrace{\mathbf{Q}^{(i)}}_{\mathbb{R}^{\overline{m}\times\overline{m}}}\underbrace{(\mathbf{S}_p^{(i)})^T}_{\mathbb{R}^{\overline{m}\times p}},$$

where $\odot$ is the (element-wise) Hadamard product.

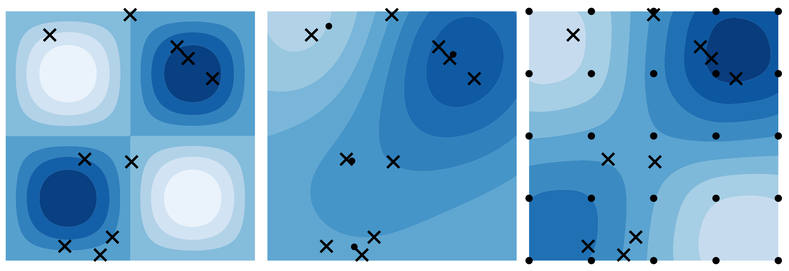
Using this result:

- Time complexity decreases from $\mathcal{O}(\overline{m}^d np) \rightarrow \mathcal{O}(dnp)$, and
- Storage complexity decreases from $\mathcal{O}(\overline{m}^d n) \rightarrow \mathcal{O}(np)$.

Note that we find $\mathbf{K}_{\mathrm{X,U}}$ admits a row-partitioned Khatri-Rao product ($*$) structure

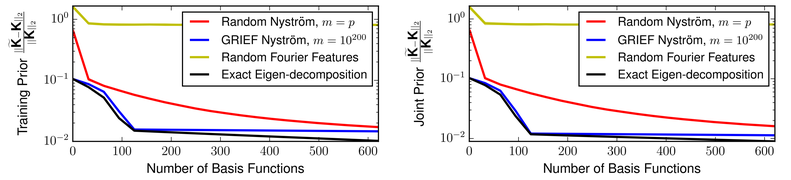
$$\mathbf{K}_{\mathrm{X,U}} = \underset{i=1}{\overset{d}{*}}\mathbf{K}_{\mathrm{X,U}}^{(i)} = \begin{pmatrix} \mathbf{K}_{\mathrm{X,U}}^{(1)}(1,:)\otimes\mathbf{K}_{\mathrm{X,U}}^{(2)}(1,:)\otimes\cdots\otimes\mathbf{K}_{\mathrm{X,U}}^{(d)}(1,:) \\ \mathbf{K}_{\mathrm{X,U}}^{(1)}(2,:)\otimes\mathbf{K}_{\mathrm{X,U}}^{(2)}(2,:)\otimes\cdots\otimes\mathbf{K}_{\mathrm{X,U}}^{(d)}(2,:) \\ \vdots \quad\quad \vdots \quad\quad \ddots \quad\quad \vdots \\ \mathbf{K}_{\mathrm{X,U}}^{(1)}(n,:)\otimes\mathbf{K}_{\mathrm{X,U}}^{(2)}(n,:)\otimes\cdots\otimes\mathbf{K}_{\mathrm{X,U}}^{(d)}(n,:) \end{pmatrix}, \tag{1}$$

and that $\mathbf{S}_p^T$ can be written as a column-partitioned Khatri-Rao product matrix. We can now estimate the kernel hyperparameters with the complexity $\mathcal{O}(np^2 + dnp + dm^{3/d})$ which is practically *independent* of $m$!



(a) Test Data. $f(x,y)=\sin(x)\sin(y)$ (b) VFE $m = 4$. RMSE $= 0.47$ (c) GP-GRIEF $m$=25, $p$=4. RMSE $= 0.34$

Figure: Reconstruction using GP-GRIEF outperforms VFE. Both techniques use an equal number of basis functions and have the same computational complexity. Crosses denote training point positions and dots denote inducing point locations. In fact, GP-GRIEF matches the test error of an exact GP.

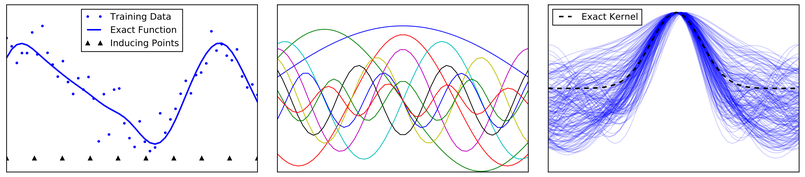



(a) Training prior covariance error. (b) Train/test joint prior covariance error.

Figure: Covariance matrix reconstruction error of GP-GRIEF outperforms competing approximation techniques. GP-GRIEF approaches the optimal reconstruction accuracy of the black curves.

## Type-I Inference in $\mathcal{O}(p)$

Consider the kernel parameterization $\widetilde{k}(\mathbf{x},\mathbf{z}) = \sum_{i=1}^{p} w_i\phi_i(\mathbf{x})\phi_i(\mathbf{z})$. If the eigenfunctions $\phi_i$ are fixed, we can compute the log marginal likelihood in $\mathcal{O}(p)$ and still approximately recover a wide class of kernels. The complexity will be independent of dataset size, and fast iterations allow type-I Bayesian inference on massive datasets.




Choose inducing point grid. Compute grid-structured eigenfunctions. Run MCMC in $\mathcal{O}(p)$. Shown are posterior samples of the kernel.

Figure: One-dimensional regression example demonstrating the GP-GRIEF type-I inference procedure in $\mathcal{O}(p)$. The posterior samples of the kernel demonstrate the flexibility of this parameterization.

## UCI Regression Datasets

We present performance benchmarks on large UCI regression datasets. Observe that:

- Complexity independence on $m$ enables use of $10^{33}$ inducing points on *Cancer*.
- GP-GRIEF-I using the developed $\mathcal{O}(p)$ procedure takes just 25 minutes to train on the two-million point dataset *Electric* and greatly outperforms [4]. This size is typically prohibitive for a fully Bayesian type-I approach.

| Dataset | $n$ | $d$ | $m=\overline{m}^d$ | Time (hrs) | GP-GRIEF-II | Time (hrs) | GP-GRIEF-I | Yang et al. [4] |
|---|---|---|---|---|---|---|---|---|
| cancer | 194 | 33 | $10^{33}$ | 0.007 | $\mathbf{27.843 \pm 3.910}$ | 0.667 | $30.568 \pm 3.340$ | $35 \pm 4$ |
| kin40k | 40K | 8 | $10^{8}$ | 0.38 | $\mathbf{0.206 \pm 0.004}$ | 0.649 | $\mathbf{0.206 \pm 0.004}$ | $0.28 \pm 0.01$ |
| electric | 2M | 11 | $10^{11}$ | 8.019 | $0.064 \pm 0.002$ | 0.418 | $\mathbf{0.058 \pm 0.006}$ | $0.12 \pm 0.12$ |

Table: Mean and standard deviation of test error and average training time from 10-fold cross validation on UCI regression datasets. GP-GRIEF-II uses maximum likelihood estimates of the hyperparameters for type-II inference whereas GP-GRIEF-I uses a fully Bayesian type-I approach using MCMC. We compare our results with Yang et al. [4] who uses the same train test splits and approximates the same kernel using Fastfood finite basis function expansions. $m$ is the number of inducing points used and $p$ is the number of eigenfunctions used.

**Acknowledgements:** Work funded by an NSERC Discovery Grant and the Canada Research Chairs program.

[1] H. Peng and Y. Qi. "EigenGP: Gaussian Process Models with Adaptive Eigenfunctions." In: *International Joint Conference on Artificial Intelligence*. 2015, pp. 3763–3769.

[2] C. T. H. Baker. *The numerical treatment of integral equations*. Oxford: Clarendon press, 1977.

[3] A. G. Wilson and H. Nickisch. "Kernel Interpolation for Scalable Structured Gaussian Processes (KISS-GP)". In: *International Conference on Machine Learning*. 2015, pp. 1775–1784.

[4] Z. Yang, A. J. Smola, L. Song, and A. G. Wilson. "À la Carte – Learning Fast Kernels". In: *Artificial Intelligence and Statistics*. 2015, pp. 1098–1106.