# Randomly Projected Additive Gaussian Processes for Regression

**Ian Delbridge** [1] **David Bindel** [1] **Andrew Gordon Wilson** [2]

## Abstract

Gaussian processes (GPs) provide flexible distributions over functions, with inductive biases controlled by a kernel. However, in many applications Gaussian processes can struggle with even moderate input dimensionality. Learning a low dimensional projection can help alleviate this curse of dimensionality, but introduces many trainable hyperparameters, which can be cumbersome, especially in the small data regime. We use additive sums of kernels for GP regression, where each kernel operates on a different random projection of its inputs. Surprisingly, we find that as the number of random projections increases, the predictive performance of this approach quickly converges to the performance of a kernel operating on the original full dimensional inputs, over a wide range of data sets, *even if we are projecting into a single dimension*. As a consequence, many problems can remarkably be reduced to one dimensional input spaces, without learning a transformation. We prove this convergence and its rate, and additionally propose a deterministic approach that converges more quickly than purely random projections. Moreover, we demonstrate our approach can achieve faster inference and improved predictive accuracy for high-dimensional inputs compared to kernels in the original input space.

## 1. Introduction

Gaussian processes (GPs) are flexible Bayesian nonparametric models with well-calibrated predictive uncertainties. Gaussian processes can also naturally encode inductive biases, such as smoothness or periodicity, through a choice of kernel function (Rasmussen and Williams, 2006). Gaussian processes have been especially impactful in the small-data regime, where careful uncertainty representation is particularly crucial, strong priors provide useful biases where learning is difficult, and exact inference is tractable. Additionally, Gaussian processes have been most successfully applied to low-dimensional input (predictor) spaces, such as time series, and spatiotemporal regression problems (e.g., Wilson and Adams, 2013; Duvenaud, 2014; Herlands et al., 2019). In these settings, canonical kernels — such as the RBF or Matérn kernels — provide reasonable similarity measures over pairs of data instances; for example, if we are modelling $CO_2$ concentrations indexed by time, then $CO_2$ levels at times which are close together in $\ell_2$ or $\ell_1$ distance will be treated as highly correlated under these kernels.

For higher dimensional problems, these standard distance metrics become less compelling. For example, with an RBF kernel, the fraction of data space with high covariance with a given point decreases exponentially with dimension. Additionally, in many online settings where Gaussian processes are used as regression models, such as Bayesian optimization, there is exponential regret with dimensionality (Srinivas et al., 2010; Bull, 2011). Furthermore, scalable Gaussian processes which have a high degree of accuracy often only apply for up to a few input dimensions (e.g., Wilson and Nickisch, 2015; Gilboa et al., 2013).

To help circumvent such issues, there are two popular approaches. The first approach is to *learn* a projection into a lower dimensional space, such as through deep kernel learning (Wilson et al., 2016). While such approaches are highly flexible, they introduce many hyperparameters to train, which can be burdensome and impractical in the small data regime. Alternatively, additive Gaussian processes (Duvenaud, 2014; Kandasamy et al., 2015; Hastie and Tibshirani, 1986) instead consider a sum of kernels, with each kernel operating on subsets of the input dimensions. This structure can both help reduce the effective dimensionality of the problem, and provide a useful inductive bias with compelling sample complexity (Stone et al., 1985). However, while assuming a fully additive decomposition of an untransformed space can provide a useful inductive bias for many real data sets, it is often too restrictive (Li et al., 2016). Moreover, methods for learning additive structure, as with standard projection approaches, are either computationally expensive or require learning a large number of parameters, which may overfit or hurt uncertainty estimation.

[1]Department of Computer Science, Cornell University, Ithaca, New York, USA [2]Center for Data Science, New York University, New York City, New York, USA. Correspondence to: Ian Delbridge <iad35@cornell.edu>.

*Proceedings of the $37^{th}$ International Conference on Machine Learning*, Vienna, Austria, PMLR 119, 2020. 

In this work, we show how to dramatically reduce the input dimensionality of a given problem, while retaining or even improving predictive accuracy, without having to learn projections. Specifically, our contributions are as follows:

- We propose a novel learning-free algorithm for constructing additive GPs based on sequences of multiple random projections (RPA-GP). This results in a Turning Band style (Matheron, 1973) approximation to a high-dimensional kernel.
- We prove that RPA-GP converges to a full-degree inverse multiquadratic kernel as the number of projections increase at a rate of $\mathcal{O}(J^{-1/2})$ where $J$ is the number of projections.
- We propose a deterministic algorithm (DPA-GP) to minimize projection redundancy and achieve faster convergence to the limiting kernel.
- We demonstrate the surprising result that RPA-GP and DPA-GP converge very quickly to the regression accuracy of a kernel operating on the full dimensional inputs, over a wide range of regression problems, *even for projections into a single input dimension*.
- We show in a large empirical study that fully additive GPs can also perform competitively with GPs using standard kernels, but are outperformed by DPA-GP with automatic relevance determination on the original input space, particularly on large data sets and high dimensional data sets.
- We additionally demonstrate that by exploiting the additive structure of RPA-GP, we alleviate the curse of dimensionality computationally for structured kernel interpolation (SKI) (Wilson and Nickisch, 2015), enabling linear-time training and constant-time predictions over a wide range of problems, including problems with over 1000 input dimensions.
- We provide GPyTorch (Gardner et al., 2018) code at `https://github.com/idelbrid/Randomly-Projected-Additive-GPs`.

The high level idea of random projections to compose additive kernels has been considered in geostatistics under the name the *turning band method* (TBM) (Matheron, 1973), for 2 and 3-dimensional *simulation*. However, the execution and details are very different from what we consider here. This paper analyzes and demonstrates how learning-free additive projections can be promising for *regression* in high dimensional input spaces.

RPA-GP and DPA-GP are a step towards alleviating the computational and analytical difficulties of high-dimensionality for Gaussian processes, while retaining a pleasingly tractable and lightweight representation. We focus our experiments on regression, since regression is the basic foundation for many popular procedures involving Gaussian processes, such as Bayesian optimization (Močkus, 1975), and model based reinforcement learning (Deisenroth and Rasmussen, 2011; Engel et al., 2005), and is in itself a widespread application for Gaussian processes (Williams and Rasmussen, 1996; van Beers and Kleijnen, 2004).

## 2. Background

We briefly review Gaussian process regression and structured kernel interpolation (SKI) (Wilson and Nickisch, 2015). For more details on Gaussian processes, we refer the reader to Rasmussen and Williams (2006).

### 2.1. Gaussian process regression

Formally, a Gaussian process $f$ is a stochastic process over an index set $\mathcal{X}$ (typically elements of $\mathcal{X}$ are in $\mathbb{R}^d$) taking on real values. Therefore, it can be interpreted as a prior over functions from $\mathcal{X}$ to $\mathbb{R}$. The process evaluated at any finite collection of points is distributed according to a multivariate normal distribution. That is, for any $\boldsymbol{x}_1, ..., \boldsymbol{x}_n \in \mathcal{X}$, $\boldsymbol{f} = [f(\boldsymbol{x}_1), ..., f(\boldsymbol{x}_n)] \sim \mathcal{N}(\boldsymbol{m}_X, K_{X,X})$. Accordingly, a Gaussian process is fully determined by its prior mean function $m : \mathcal{X} \mapsto \mathbb{R}$ and covariance kernel function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$. The prior mean function is often chosen to be 0 in the case where we have limited knowledge of $f$. Therefore, a Gaussian process is almost entirely determined by $k$. Standard identities of the multivariate Gaussian distribution can be applied to find the posterior predictive distribution under a Gaussian observation model given data $X, \boldsymbol{y} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ at points $X_*$ is

$$\boldsymbol{f}_* | X, \boldsymbol{y}, X_* \sim \mathcal{N}(\bar{\boldsymbol{f}}_*, \mathrm{cov}(\boldsymbol{f}_*)),$$

where

$$\bar{\boldsymbol{f}}_* := K_{X_*,X}(K_{X,X} + \sigma^2 I_n)^{-1}\boldsymbol{y},$$
$$\mathrm{cov}(\boldsymbol{f}_*) := K_{X_*,X_*} - K_{X_*,X}(K_{X,X} + \sigma^2 I_n)^{-1}K_{X,X_*}.$$

The computational bottleneck in computing the posterior distribution is solving the linear system $(K_{X,X}+\sigma^2 I_n)^{-1}\boldsymbol{y}$. Standard approaches use the Cholesky decomposition, which requires $\mathcal{O}(n^3)$ computations.

The log marginal likelihood

$$\log p(\boldsymbol{y}|X) = -\frac{1}{2}\boldsymbol{y}^\top (K_{X,X} + \sigma^2 I_n)^{-1}\boldsymbol{y} - \frac{1}{2}\log |K_{X,X} + \sigma^2 I_n| - \frac{n}{2}\log 2\pi$$

is used for model comparison and optimization. Typically, one parameterizes the kernel with some number of hyperparameters which are tuned by maximizing the marginal

likelihood. This maximization provides automatic regularization because the determinant $|K_{X,X} + \sigma^2 I_n|$ penalizes quickly varying functions. The computational bottleneck in computing the marginal likelihood is the determinant, which has the standard computational cost of $\mathcal{O}(n^3)$ from the Cholesky decomposition.

### 2.2. Structured kernel interpolation

Structured kernel interpolation (SKI) (Wilson and Nickisch, 2015) uses an approximation to the kernel $K_{X,X}$ that permits fast matrix-vector multiplications, which are used to compute the log marginal likelihood and predictive distributions. Specifically, let $U$ be a regular grid of inducing points. Wilson and Nickisch (2015) let $W$ be a matrix of interpolation weights from $U$ to $X$ determined by the product of local cubic interpolation weights of the nearest four inducing points in each dimension (Keys, 1981). The SKI approximation of base kernel matrix $K_{X,X}$ is

$$K_{X,X} \approx K^{\text{SKI}}_{X,X} := W K_{U,U} W^\top.$$

The interpolation matrix $W$ is sparse, having only $4^d$ nonzero elements per row. The matrix $K_{U,U}$ can have Toeplitz (if $d = 1$) or Kronecker (if $d > 1$) structure, either of which permit fast matrix-vector multiplications with $K_{U,U}$ (Saatçi, 2012) and thus also the approximate $K_{X,X}$, due to the sparse interpolation in SKI. The linear solve $(K^{\text{SKI}}_{X,X} + \sigma^2 I_n)^{-1}\boldsymbol{y}$ can then be efficiently computed using linear conjugate gradients, which proceeds by iterative matrix-vector multiplications of $K^{\text{SKI}}_{X,X} + \sigma^2 I_n$. The log determinant can be computed using stochastic Lanczos quadrature (Dong et al., 2017), which similarly only requires iterative matrix-vector multiplications of $K^{\text{SKI}}_{X,X} + \sigma^2 I_n$.

However, fixing the number of inducing points in each dimension, the size of the grid grows exponentially with dimension. Therefore, inference using SKI is intractable generally for dimension $d > 5$.

## 3. Related Work

GPs with kernels that fully decompose additively[1], i.e.

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{d} k_i(x_i, x_i'), \tag{1}$$

for some sub-kernels $\{k_i\}_{i=1}^d$, are considered "Generalized Additive Models" (GAM) (Hastie and Tibshirani, 1986) or "Additive Kriging Models" (AKM) (Durrande et al., 2012). We refer to the resulting GP as a GAM GP throughout this paper. Here, we denote the $i$th component of vector $\boldsymbol{x}$ as $x_i$ without bold face to indicate that it is a scalar value.

The GAM GP implicitly assumes there are only first-order interactions in the modeled function. This assumption may be reasonable inductive bias in some cases; indeed Durrande et al. (2012) make a strong case for using GAM GPs. However, the assumption that there are no interactions between input dimensions is often too strong (Li et al., 2016; Duvenaud et al., 2011). It is natural, then, to consider additive combinations of sets of features. Unfortunately, the space of subsets of features is a power set and therefore grows exponentially. Therefore, learning additive combinations of kernels on subsets of features is difficult. Extant approaches to learning additive kernel structure can be divided roughly into *enumeration* methods, where sub-kernels consider every combination of feature interactions up to a degree, *search methods*, where possible decompositions are traversed by a search algorithm, and *projection-pursuit* where a projected-additive GP is learned by iteratively optimizing projection directions from regression residuals.

Hierarchical Kernel Learning (Bach, 2009) is an enumeration method in which one constructs the sum of kernels in a hull of possible kernels. Duvenaud et al. (2011) compute the sum of kernels over every possible feature combination in $O(d^2)$ time by using the Newton-Girard formulae. Duvenaud et al. (2013) define a grammar over kernels and use discrete search to optimize kernel structure. Qamar and Tokdar (2014) uses a sampling approach to search through additive decompositions. In Bayesian optimization, it is especially beneficial to learn additive structure where no features are overlapped between sub-kernels. Gardner et al. (2017) perform MCMC sampling over such kernel structures as a search method. Similarly, Wang et al. (2017) perform a Gibbs sampling procedure to search over feature partitions. Enumeration methods inherently incorporate a very large number of sub-kernels, which can be expensive to compute for high dimensions. Search methods, on the other hand, are burdened by searching over a combinatorial space.

Projection pursuit, introduced by Friedman and Stuetzle (1981) and adapted to the Gaussian process setting by Saatçi (2012); Gilboa et al. (2013), is different in that one learns *projected*-additive GPs. That is, the GP kernel is an additive combination of low-dimensional kernels defined on linear projections of data whose directions are sequentially optimized. If a large number of projections are used, the sequential optimization of directions with respect to the marginal likelihood can be computationally expensive, and the large number of parameters learned by optimization may result in overfitting and poor uncertainty estimation (Li et al., 2016).

A GP using a single non-additive random projection has

[1] We define the *degree* of a kernel to be the number of dimensions over which it operates. We say a kernel is *additive* simply if it is a sum of lower-degree kernels. Moreover, a GP is additive if its kernel is additive.

been briefly considered with promising preliminary results (Wang et al., 2016). However, we find that such methods can be dramatically improved through sequences of additive random and deterministic projections, and investigate this surprising and practically significant result. Additionally, Guhaniyogi and Dunson (2016) use a GP over random projections of high-dimensional data having low-dimensional manifold structure. However, their work does not explore additive Gaussian processes and also relies on a model average of many GPs to account for variation in the random projections. These methods are also related to *multiple-index models* (Xia, 2008), which are a general class of models of the form $G(Px) + \beta^\top x + \epsilon$ where $P$ is a parameter matrix, $\beta$ is a parameter vector, and $G$ is an unknown link function, though $P$ is typically learned.

Composing 1-dimensional stochastic processes along random directions to approximate higher-dimensional stochastic processes has been used in the geostatistics community under the name the "turning bands method" and has since been studied in detail for simulation of 2 to 3-dimensional processes (Matheron, 1973; Mantoglou and Wilson, 1982; Mantoglou, 1987; Lantuéjoul, 2013). Work has been devoted to describing the 1-dimensional covariances associated with common covariances (Christakos, 1987; Gneiting, 1998), quantifying the approximation error (Mantoglou, 1987), and even choosing well-spread directions (Freulon and Lantuejoul, 1993; Lantuéjoul, 2013). Yet, this direction of work has not been explored for higher dimensional GPs, nor for Gaussian process regression.

## 4. Randomly Projected-Additive GPs

Rather than directly learning additive structure, we project data onto randomly drawn directions and impose additive structure on a GP defined over the projections. As a result, we bypass the need to search over or enumerate all possible sub-kernels, and the burden of training many hyperparameters in a learned projection.

Formally, let $n$ be the number of data points, $d$ be number of dimensions, and $J$ be the number sub-kernels. Denoting the degree of kernel $j$ as $D_j$, we define the randomly projected additive kernel as

$$k_{rp}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{j=1}^{J} \alpha_j k_j(P^{(j)}\boldsymbol{x}, P^{(j)}\boldsymbol{x}'), \tag{2}$$

$$\forall j \in [J], \quad P^{(j)} \in \mathbb{R}^{D_j \times d}, \tag{3}$$

$$P^{(j)}_{r,c} \sim_{\text{i.i.d.}} \mathcal{N}\left(0, \frac{1}{D_j}\right) \quad \forall r \in [D_j], c \in [d]. \tag{4}$$

We refer to a GP with covariance kernel $k_{rp}$ as a randomly-projected additive GP (RPA-GP). Matrices $\{P^{(j)}\}_{j=1}^J$ define the directions of the projections. The parameters

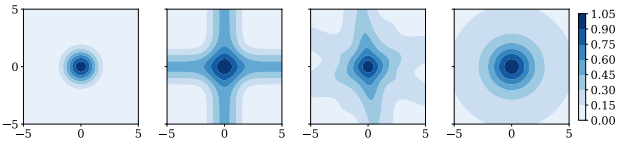


*Figure 1.* Contour plots of 2-dimensional kernels. From left to right: RBF, GAM RBF, RPA-GP with 16 projections, and DPA-GP with 16 projections. With enough additive projections, we attain approximately spherical covariance, and choosing well-placed directions facilitates convergence.

$\{\alpha_j\}_{j=1}^J$ determine the amount of variance each sub-kernel contributes and may be either learned or set as a constant value $1/J$.

If the sub-kernel degrees $D_j$ are large enough relative to sample size $n$, the Johnson-Lindenstrauss Lemma guarantees that the $\ell_2$ distances between points are approximately preserved with high probability (Sarlos, 2006). Alternatively, we have a similar guarantee if data lie on a low-dimensional manifold (Baraniuk and Wakin, 2009). Therefore, if we use RBF sub-kernels, each sub-kernel is a good approximation of the high-dimensional RBF kernel. Moreover, having multiple random projections increases the likelihood of drawing a good random projection but does not increase kernel dimensionality, similar to the method presented in Ahmed (2004). However, if the sub-kernel degrees are small, we have sufficient flexibility given enough projections. RPA-GP forms a distribution over linear combinations of *ridge functions*, which are defined as functions that are invariant in all but 1 direction. Since linear combinations of ridge functions are dense in the set of continuous functions, we are able to approximate any continuous function arbitrarily well given a rich enough set of directions (Cheney and Light, 2009).

### 4.1. The expected kernel

We now analyze projected-additive kernels by studying the functional form of the covariance in comparison to the RBF kernel. We limit our analysis to the most challenging case of one-dimensional additive projections, i.e. when each matrix $P^{(j)}$ is a vector $\boldsymbol{\eta}_j$, though analysis is similar for higher dimensional projections. Further, we assume unit length-scale, which can always be achieved by appropriate scaling of data. For brevity, we defer proofs to the appendix.

Clearly, an additive (GAM) covariance kernel does not decay to zero as the $\ell_2$ distance between points goes to infinity. For example, a GAM kernel with RBF additive components is lower bounded by $\frac{d-1}{d}$ along each axis. Conversely, in expectation, the covariance of a randomly projected additive kernel with RBF components decays to zero in any given

direction; if $\boldsymbol{\eta}_j$ are drawn from an isotropic distribution, an additive randomly projected kernel converges to a high-dimensional kernel as $J \to \infty$. This is made formal in the following proposition.

**Proposition 1.** *Let $\phi\colon \mathbb{R} \mapsto [-1, 1]$ be a 1-dimensional kernel, and let $(\boldsymbol{\eta}_j\colon j \geq 1)$ be an i.i.d. sequence of random vector in $\mathbb{R}^d$ drawn from an isotropic distribution. Then, for some expected kernel $k_e\colon \mathbb{R} \mapsto [-1, 1]$, for any $\boldsymbol{\tau} \in \mathbb{R}^d$, almost surely*

$$\lim_{J\to\infty} \frac{1}{J} \sum_{j=1}^{J} \phi(\boldsymbol{\eta}_j^\top \boldsymbol{\tau}) = \mathbb{E}[\phi(\eta_{11}||\boldsymbol{\tau}||_2)] =: k_e(||\boldsymbol{\tau}||_2).$$

**Corollary 1.** *If $\phi(x) = e^{-\frac{1}{2}x^2}$ and $\boldsymbol{\eta}_1 \sim \mathcal{N}(0, I_d)$, then*

$$k_e(\boldsymbol{\tau}) = \frac{1}{\sqrt{1 + ||\boldsymbol{\tau}||_2^2}} \triangleq k_{IMQ}(\boldsymbol{\tau}). \tag{5}$$

**Corollary 2.** *If $\phi(x) = \cos(x)$ and $\boldsymbol{\eta}_1 \sim \mathcal{N}(0, I_d)$, then*

$$k_e(\boldsymbol{\tau}) = e^{-\frac{1}{2}||\boldsymbol{\tau}||_2^2} \triangleq k_{RBF}(\boldsymbol{\tau}). \tag{6}$$

Corollaries (1) and (2) show that for certain choices of sub-kernel $\phi$ and projection distribution, we can recover familiar kernels. The expected kernel in (5) is a rational quadratic kernel with parameter $\alpha = 1/2$, also known as the *inverse multiquadratic kernel*. It is especially relevant because in this work we focus on the case when the base kernel is RBF. Note that the spectral density provides a standard way to derive sub-kernels associated with higher-dimensional kernels (Mantoglou, 1987). We can also derive a convergence rate.

**Proposition 2.** *Let $\phi$, $k_e$, $(\eta_j\colon j \geq 1)$ be as in Proposition 1. Let $\delta > 0$. Then, with probability at least $1 - \delta$, we have simultaneously for all pairs of points $\boldsymbol{\tau}_{i,k}$, $i, k \in [n]$,*

$$\left|\frac{1}{J}\sum_{j=1}^{J}\phi(\boldsymbol{\eta}_j^\top \boldsymbol{\tau}_{i,k}) - k_e(||\boldsymbol{\tau}_{i,k}||_2)\right| \leq \frac{4}{3J}\log(n^2/\delta) + \sqrt{\frac{2\sup_{i,k}\operatorname{var}(\phi(\boldsymbol{\eta}_1^\top \boldsymbol{\tau}_{i,k}))}{J}\log(n^2/\delta)}$$

Empirically, we see convergence to the performance of a kernel operating on the original space at a much greater rate. This empirical result is intuitive because even if the resulting kernel after additive random projections is not an inverse multiquadratic kernel, it may still be a good kernel for the data.

### 4.2. Reducing projection redundancy

As shown in Proposition 2, sampling directions purely randomly converges at the "slow" simple Monte Carlo rate of $\mathcal{O}(1/\sqrt{J})$. Ideally, we would space directions equally. However, even in only 3-dimensions, this is only possible for certain values of $J = 3, 15, ...$ (Mantoglou, 1987). In higher dimensions, the problem is highly nontrivial. One solution is to numerically maximize a measure of distance between points, such as the antipodal separation distance

$$\delta(\boldsymbol{\eta}_1, ..., \boldsymbol{\eta}_J) = \min_{j\neq j'} \cos^{-1}(|\boldsymbol{\eta}_j^\top \boldsymbol{\eta}_{j'}|),$$

which directly measures the minimal angle between directions. However, because maximizing $\delta$ is difficult, we instead minimize the loss

$$\ell(\boldsymbol{\eta}_1, ..., \boldsymbol{\eta}_J) = \sum_{j\neq j'} (\boldsymbol{\eta}_j^\top \boldsymbol{\eta}_{j'})^4. \tag{7}$$

Minimizing $\ell$ has the effect of increasing the separation distance $\delta$ between directions, though an optimizer of $\ell$ does not necessarily coincide with an optimizer of $\delta$ unless $d \geq J$. Additionally, given sufficiently large $J$, a set of directions $\{\boldsymbol{\eta}_j\}_{j=1}^J$ that maximize $\ell$ is a spherical $t$-design with $t = 4$ (Womersley, 2018), thus guaranteeing *optimal order rate decay of worst-case error* for quadrature of smooth functions. If $J \leq d$, orthogonal directions minimize $\delta$ and $\ell$, so we simply use Gram-Schmidt orthogonalization. Otherwise, we minimize $\ell$ using gradient descent. We refer to projected-additive GPs with directions chosen by this method as Diverse Projected-Additive GPs (DPA-GP). We visualize the DPA-GP and other kernels in Figure 1.

### 4.3. Applying length-scales before projection

If each sub-kernel of an additive kernel learns its own length-scale, it is not clear that the additive kernel approximates an expected kernel. Additionally, if the number of additive kernels $J$ is large, learning separate length-scales introduces many hyperparameters which are also only indirectly related to the original inputs.

Alternatively, we propose applying automatic relevance determination scaling *directly on the original input space* before the data are projected to a low-dimensional space. To learn the length scales, we efficiently propagate gradients through the projections with automatic differentiation. We define

$$k_{rpARD}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{j=1}^{J} \alpha_j k_j(P^{(j)}A\boldsymbol{x}, P^{(j)}A\boldsymbol{x}'),$$

where $A = \operatorname{diag}(\boldsymbol{\sigma}^{-1})$. When $\alpha_j = 1/J$ for all $j$ and each $k_j$ has unit length-scales, the theory of section 4.1 readily applies, while permitting flexible treatment of length-scales.

In Section 5, we make the empirical discovery that this ARD approach provides significant performance gains. To distinguish this parameterization from others, we refer to such a model with the *-ARD* suffix.

### 4.4. Scaling to large data sets with high dimension with SKI

In section 2.2, we described how SKI enables scalable GPs, but is constrained to input dimensions of about $d < 5$, if no kernel structure is exploited. However, if a kernel decomposes additively by groups of dimensions, it is possible to generalize the applicability of SKI to much higher dimensional spaces. Suppose that a kernel decomposes additively, as in the case of RPA or DPA, i.e.

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{j=1}^{J} k_j(\boldsymbol{x}^{(j)}, \boldsymbol{x}'^{(j)}),$$

where $\boldsymbol{x}^{(j)}$ denotes a group of dimensions of point $\boldsymbol{x}$. Then, the Gram matrix $K$ corresponding to this kernel decomposes similarly, so a matrix-vector multiplication can be performed with each kernel separately as $K\boldsymbol{v} = \sum_{j=1}^{J} K^{(j)}\boldsymbol{v}$. Since we assume that each sub-kernel $k_j$ is low-degree, each matrix-vector multiplication $K^{(j)}\boldsymbol{v}$ can be computed efficiently using SKI. In particular, in the case that each sub-kernel is 1-dimensional, inference with such a kernel using SKI has complexity $\mathcal{O}(Jc(n + m \log m))$, where $m$ is the number of inducing points for each projection and $c$ is the number of iterations of linear conjugate gradients. Typically, $c \ll n$ is sufficient to reach convergence within machine precision, so inference is approximately linear in $n$ (Wilson and Nickisch, 2015). Moreover, MVMs can be trivially parallelized across additive components. We demonstrate this asymptotic scaling empirically in Section 5.5.

## 5. Experiments

We evaluate RPA-GP and DPA-GP on a wide array of regression tasks. We compare the predictive accuracy of the proposed methods to GPs with RBF and GAM kernels (section 5.1), study the effect of increasing number of projections (section 5.2), address the choice of the number of projections (5.3), compare predictive accuracy under various assumptions via synthetic data sets and on very high-dimensional data (section 5.4), and demonstrate the superior asymptotic scaling of RPA-GP with SKI over traditional inference (section 5.5). We implement all models using GPyTorch (Gardner et al., 2018) and provide code at `https://github.com/idelbrid/Randomly-Projected-Additive-GPs`.

We note that the most directly relevant comparison is to GP with a kernel operating on the original input space, such as the popular RBF-ARD kernel. *Kernel learning* approaches have a completely different purpose. For example, deep kernel learning (Wilson et al., 2016) trains a deep projection to perform kernel selection, requiring the training of many hyperparameters, which is cumbersome and even infeasible for many smaller datasets. On the other hand, the remarkable feature of our proposed approaches is that they provide *learning-free* projections into a single dimension without sacrificing accuracy compared to a popular kernel operating on the original input space.

### 5.1. Benchmarks on UCI data sets

To evaluate RPA and DPA-GP, we compute the normalized RMSE and negative log likelihood for a large number of UCI data sets. For full details of the experiment procedure, results for additional models, and the negative log likelihood (NLL) benchmarks, refer to Appendix B. Notably, using SKI for scalable inference, enabled by additive projections, results in essentially identical performance as exact inference. Additionally, the NLL for DPA-GP-ARD is also competitive, indicating that predictive uncertainty is not worsened by additive random projections.

To reiterate, `GAM-GP` is an additive GP with RBF subkernels and ARD. `RPA-GP-1`, `DPA-GP`, and `DPA-GP-ARD` are additive across 20 1-dimensional projections. `RPA-GP-1` uses Gaussian random projections; `DPA-GP` minimizes the objective in Equation 7; and `DPA-GP-ARD` performs the pre-projection ARD method described in Section 4.3.

### 5.2. Convergence of random projected-additive GP accuracy

To study the sensitivity of RPA-GP to the number of projections and limiting behavior as $J \to \infty$, we measure the RMSE of projected-additive GPs as the number of projections vary. We show representative plots in Figure 3. In these experiments, the benefit of DPA-GP-ARD also becomes obvious, as we see it is able to converge much more quickly than the other approaches. By the time $J = 20$, represented in Figure 2, the various projection approaches are more comparable.

### 5.3. Choosing the number of projections $J$

One can often achieve good results with a surprisingly small value of $J$. Indeed, in Figure 2, we fixed $J = 20$, and found that methods projecting into a single dimension often met or exceeded the performance of kernels operating on the full input space, over a wide range of datasets with varying numbers of points and input dimensions.

We also show in Figure 3 that convergence to a good solution can be very fast in $J$, especially with DPA-GP. In general, $J$ can be determined through cross-validation. Alternatively, one may refer to Proposition 2 to choose $J$ such that the projected additive kernel is with in $\epsilon$ of its expected kernel. However, these values are generally overly conservative, as a reasonable kernel for a given dataset will be found much more quickly than a close approximation to the IMQ kernel.

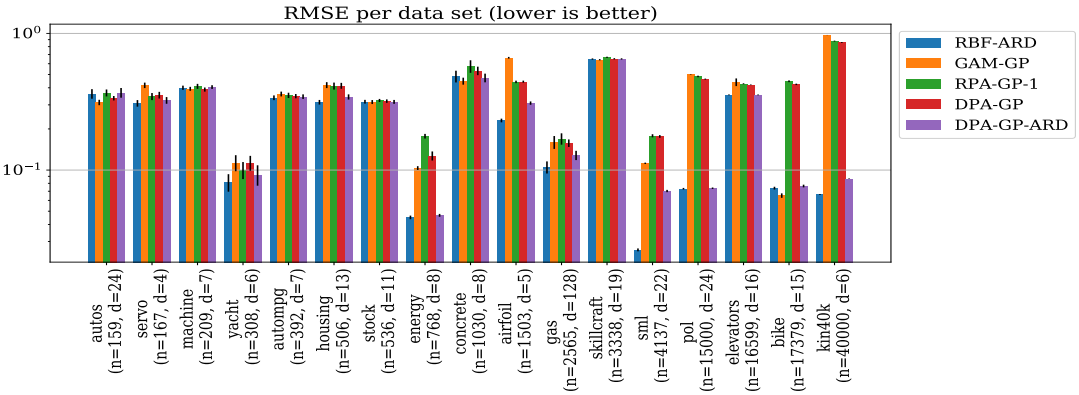


*Figure 2.* The average RMSE and SEMs of the proposed methods. The proposed methods and GAM-GP perform surprisingly well compared to RBF-GP. DPA-GP-ARD is able to match the performance of RBF-GP even for large data sets, where the flexibility of GAM-GP begins to be a limiting factor. Table 2 in Appendix B additionally compares predictive log likelihood, showing a similar trend.

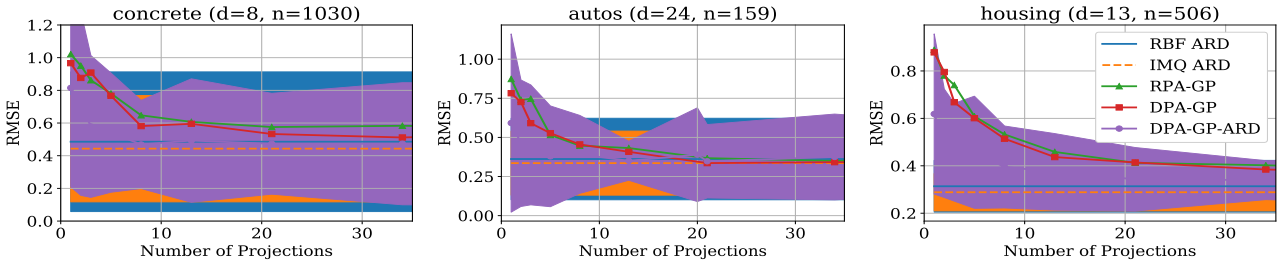


*Figure 3.* Representative test RMSE of RPA-GP and DPA-GP as the number of projections vary compared to full-dimensional RBF and inverse multiquadratic (IMQ) kernels. Shaded regions are 2 times the standard deviation over cross-validation, and lines are the average RMSE. For clarity, we only show the variation for DPA-GP-ARD. In general, there is a fast convergence to the performance of RBF and IMQ kernels, and DPA-GP consistently improves upon RPA-GP by a small amount, and applying length-scales before (DPA-GP-ARD) projection dramatically increases performance.

Indeed, with smaller data sets ($n < 3000$), DPA-ARD empirically attains RMSE within five percent of RBF-ARD for $J < 20$, even if the dimensionality of the dataset is quite large. For larger datasets, RMSE still is at a local minimum for $J < d$, which is a more suitable predictive performance-computational cost trade-off. Appendix C contains additional plots.

### 5.4. Comparisons to fully-additive kernel

As observed in section 5.1, GAM GP performs surprisingly well in comparison to the standard RBF-ARD kernel, despite its limited model class. This result is noteworthy in its own right, since the GAM GP ignores interactions between inputs. To our knowledge it is not known that such a parsimonious representation can achieve comparable results to a kernel acting on the full inputs over this significant range of experiments.

To understand the differences between GAM GP and our proposed techniques, we consider additional empirical tests

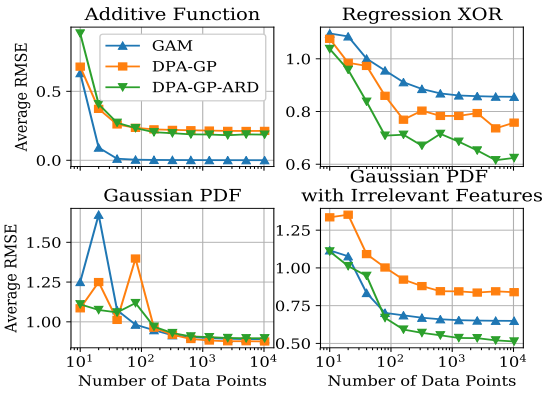


*Figure 4.* **Top left**: RMSE on a sum of sines. GAM is best as expected. **Top right**: RMSE on a very non-additive function: a smooth relaxation of the XOR function. **Bottom left**: RMSE synthetic data with a rotation-invariant function. **Bottom right**: RMSE synthetic data with half irrelevant features. DPA-GP-ARD can determine irrelevant features effectively.

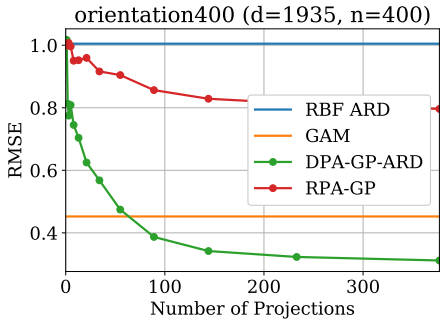


*Figure 5.* Comparative performance on the Olivetti face orientation regression task. With high dimensionality and a non-additive target function, DPA-GP-ARD outperforms alternatives, though the number of projections $J$ must be somewhat high.

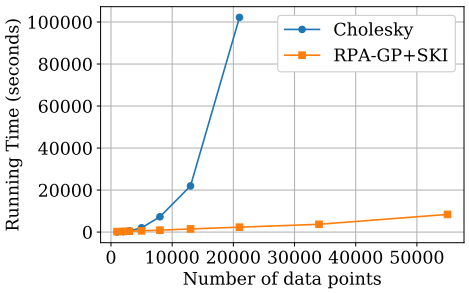


*Figure 6.* Training run time of RPA-GP with SKI compared to canonical Cholesky decomposition-based inference on synthetic data following $x \sim N(0, I_d)$, $y = \sum_{i=1}^{100} \sin(x_i) + \epsilon$. We note that runtime behaviour is somewhat data dependent, due to changes in conditioning of the kernel matrix, but the scaling of SKI remains approximately linear, compared to the cubic scaling of the Cholesky decomposition.

on synthetic data and very high-dimensional data sets.

**Synthetic regression tasks**: In Figure 4, we test each method as we increase the number of data points on additive synthetic data. GAM, as expected, performs better for target functions that are truly additive. Conversely, GAM can perform poorly if the function is not additive. For target functions that are rotation-invariant, all of GAM, DPA-GP, and DPA-GP-ARD perform equivalently as expected. However, if irrelevant features are introduced, which we conjecture is a frequent occurrence in real regression problems, DPA-GP-ARD performs better than GAM. Irrelevant features introduce noise into the projections, but ARD prunes irrelevant features, effectually reducing the input dimension.

**High dimensional regression tasks**: Following Wilson et al. (2016) and Hinton and Salakhutdinov (2008), we construct regression data sets of three different sizes from the Olivetti faces data set. We uniformly subsample images, uniformly sample a rotation in $[-90, 90]$, crop the rotated images, and use the rotations as regression targets. DPA-GP-ARD outperforms RBF and GAM GPs with when $n$ is small compared to $d$. Results with $n = 400$ images are presented in Figure 5. We additionally test on three genomics data sets, finding that DPA-GP-ARD and GAM GP generally perform comparably and provide figures in the appendix.

### 5.5. Scaling to large data sets

To demonstrate the asymptotic computational complexity of RPA-GP with SKI, we train both RPA-GP with SKI and a GP with RBF kernel using Cholesky-based inference for 120 Adam iterations on synthetic data sets with $d = 100$ and a varying number data points. We use RPA-GP with 20 1-dimensional projections and 512 inducing points per projection. We run this experiment on a 1.8 GHz Intel i5 processor and 8 GB of RAM. The results in Figure 6.

Note that it is infeasible to run SKI without RPA-GP with a reasonable number of inducing points on data sets with this number of input dimensions; even if $d = 6$ and we have 100 inducing points in each dimension, the resulting 1 trillion inducing points cannot be stored in memory.

## 6. Conclusion

We proposed novel learning-free algorithms to construct additive Gaussian processes by using sums of low-dimensional kernels operating over random (RPA-GP) projections. We demonstrated the remarkable result that these approaches achieve the performance of kernels operating over the full-dimensional input space even when projecting into a *one-dimensional* space and *without learning* the projections.

Moreover, we showed that RPA-GP converges to the inverse multiquadratic kernel and proposed a novel deterministic algorithm (DPA-GP) to reduce projection redundancy that indeed improves regression performance. Finally, as an added benefit, we demonstrated that by exploiting the additive structure of RPA-GP, we essentially reduce inference from a $d$ dimensional problem to $J$ 1-dimensional problems, enabling the application of SKI (Wilson and Nickisch, 2015) and thereby reducing standard GP computational complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(Jc(n+m))$, where $J$ is the number of random projections, $c$ is the number of linear conjugate gradients iterations, and $m$ is the number of inducing points. These results are of great practical significance: we have shown that GP regression, which is the backbone of many procedures, can often be effectively reduced to one-dimension without requiring the training of a projection. These approaches also naturally generalize the applicability of popular scalable inference procedures, such as SKI, which have been conventionally constrained to lower dimensional spaces.

In short, we demonstrate the pleasing result that a range of

regression problems can be reduced to a *single* input dimension, while retaining or even improving accuracy, without having to learn a projection. In a single dimension, methods become much easier to analyze and scale, leading to a rich variety of future research directions.

ACKNOWLEDGEMENTS

This research is supported by an Amazon Research Award, Facebook Research, Amazon Machine Learning Research Award, NSF I-DISRE 193471, NIH R01 DA048764-01A1, NSF IIS-1910266, NSF 1922658 NRT-HDR: FUTURE Foundations, Translation, and Responsibility for Data Science, and NSF DMS-1620038. We thank Alex Smola and Eytan Bakshy for helpful discussions.

## References

Yousuf Shamim Ahmed. *Multiple random projection for fast, approximate nearest neighbor search in high dimensions*. University of Toronto, 2004.

Francis Bach. High-dimensional non-linear variable selection through hierarchical kernel learning. *arXiv preprint arXiv:0909.0844*, 2009.

Richard G Baraniuk and Michael B Wakin. Random projections of smooth manifolds. *Foundations of computational mathematics*, 9(1):51–77, 2009.

Adam D Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(Oct):2879–2904, 2011.

Elliott Ward Cheney and William Allan Light. *A course in approximation theory*, volume 101. American Mathematical Soc., 2009.

George Christakos. Stochastic simulation of spatially correlated geo-processes. *Mathematical Geology*, 19(8): 807–831, 1987.

Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472, 2011.

Kun Dong, David Eriksson, Hannes Nickisch, David Bindel, and Andrew G Wilson. Scalable log determinants for gaussian process kernel learning. In *Advances in Neural Information Processing Systems*, pages 6330–6340, 2017.

Nicolas Durrande, David Ginsbourger, and Olivier Roustant. Additive covariance kernels for high-dimensional gaussian process modeling. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 21, pages 481–499, 2012.

David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

David Duvenaud, James Lloyd, Roger Grosse, Joshua Tenenbaum, and Ghahramani Zoubin. Structure discovery in nonparametric regression through compositional kernel search. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1166–1174, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL http://proceedings.mlr.press/v28/duvenaud13.html.

David K Duvenaud, Hannes Nickisch, and Carl E Rasmussen. Additive gaussian processes. In *Advances in neural information processing systems*, pages 226–234, 2011.

Yaakov Engel, Shie Mannor, and Ron Meir. Reinforcement learning with gaussian processes. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 201–208, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. doi: 10.1145/1102351.1102377. URL http://doi.acm.org/10.1145/1102351.1102377.

Xavier Freulon and Christian Lantuejoul. Revisiting the turning bands method. *Acta Stereologica*, 1993.

Jerome H Friedman and Werner Stuetzle. Projection pursuit regression. *Journal of the American statistical Association*, 76(376):817–823, 1981.

Jacob Gardner, Chuan Guo, Kilian Weinberger, Roman Garnett, and Roger Grosse. Discovering and exploiting additive structure for bayesian optimization. In *Artificial Intelligence and Statistics*, pages 1311–1319, 2017.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, pages 7576–7586, 2018.

Elad Gilboa, Yunus Saatçi, and John Cunningham. Scaling multidimensional gaussian processes using projected additive approximations. In *International Conference on Machine Learning*, pages 454–461, 2013.

Tilmann Gneiting. Closed form solutions of the two-dimensional turning bands equation. *Mathematical Geology*, 30(4):379–390, 1998.

Rajarshi Guhaniyogi and David B Dunson. Compressed gaussian process for manifold regression. *The Journal of Machine Learning Research*, 17(1):2472–2497, 2016.

Trevor Hastie and Robert Tibshirani. Generalized additive models. *Statist. Sci.*, 1(3):297–310, 08 1986. doi: 10.1214/ss/1177013604. URL https://doi.org/10.1214/ss/1177013604.

William Herlands, Daniel B. Neill, Hannes Nickisch, and Andrew Gordon Wilson. Change surfaces for expressive multidimensional changepoints and counterfactual prediction. *Journal of Machine Learning Research*, 20(99):1–51, 2019. URL http://jmlr.org/papers/v20/17-352.html.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Using deep belief nets to learn covariance kernels for gaussian processes. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1249–1256. Curran Associates, Inc., 2008.

Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning*, pages 295–304, 2015.

Robert Keys. Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing*, 29(6):1153–1160, 1981.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Christian Lantuéjoul. *Geostatistical simulation: models and algorithms*. Springer Science & Business Media, 2013.

Chun-Liang Li, Kirthevasan Kandasamy, Barnabas Poczos, and Jeff Schneider. High dimensional bayesian optimization via restricted projection pursuit models. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 884–892, Cadiz, Spain, 09–11 May 2016. PMLR. URL http://proceedings.mlr.press/v51/li16e.html.

Aristotelis Mantoglou. Digital simulation of multivariate two-and three-dimensional stochastic processes with a spectral turning bands method. *Mathematical Geology*, 19(2):129–149, 1987.

Aristotelis Mantoglou and John L Wilson. The turning bands method for simulation of random fields using line generation by a spectral method. *Water Resources Research*, 18(5):1379–1394, 1982.

G. Matheron. The intrinsic random functions and their applications. *Advances in Applied Probability*, 5(3):439–468, 1973. ISSN 00018678. URL http://www.jstor.org/stable/1425829.

Jonas Močkus. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, pages 400–404. Springer, 1975.

Shaan Qamar and Surya T Tokdar. Additive gaussian process regression. *arXiv preprint arXiv:1411.7009*, 2014.

Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*, volume 2. MIT Press Cambridge, MA, 2006.

Yunus Saatçi. *Scalable inference for structured Gaussian process models*. PhD thesis, Citeseer, 2012.

Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 143–152. IEEE, 2006.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 1015–1022, USA, 2010. Omnipress. ISBN 978-1-60558-907-7. URL http://dl.acm.org/citation.cfm?id=3104322.3104451.

Charles J Stone et al. Additive regression and other nonparametric models. *The annals of Statistics*, 13(2):689–705, 1985.

W. C. M. van Beers and J. P. C. Kleijnen. Kriging interpolation in simulation: a survey. In *Proceedings of the 2004 Winter Simulation Conference, 2004.*, volume 1, page 121, Dec 2004. doi: 10.1109/WSC.2004.1371308.

Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In *AISTATS*, 2017.

Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando de Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

Christopher KI Williams and Carl Edward Rasmussen. Gaussian processes for regression. In *Advances in neural information processing systems*, pages 514–520, 1996.

Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning*, pages 1067–1075, 2013.

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pages 1775–1784, 2015.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378, 2016.

Robert S Womersley. Efficient spherical designs with good geometric properties. In *Contemporary Computational Mathematics-A Celebration of the 80th Birthday of Ian Sloan*, pages 1243–1285. Springer, 2018.

Yingcun Xia. A multiple-index model and dimension reduction. *Journal of the American Statistical Association*, 103 (484):1631–1640, 2008.