Additive Gaussian Processes Revisited

Additive Gaussian Processes Revisited

Xiaoyu Lu [1] **Alexis Boukouvalas** [1] **James Hensman** [1]

Abstract

Gaussian Process (GP) models are a class of flexible non-parametric models that have rich representational power. By using a Gaussian process with additive structure, complex responses can be modelled whilst retaining interpretability. Previous work showed that additive Gaussian process models require high-dimensional interaction terms. We propose the orthogonal additive kernel (OAK), which imposes an orthogonality constraint on the additive functions, enabling an identifiable, low-dimensional representation of the functional relationship. We connect the OAK kernel to functional ANOVA decomposition, and show improved convergence rates for sparse computation methods. With only a small number of additive low-dimensional terms, we demonstrate the OAK model achieves similar or better predictive performance compared to black-box models, while retaining interpretability.

1. Introduction

Gaussian Processes (GPs) can be used to construct additive models by using the property that a sum of two GPs results in a new GP with a kernel function defined as the sum of the original ones. Using an additive structure in a Gaussian process model is enticing from an explainability standpoint, since one can use the linear properties of the GP to perform inference over the added components, which can yield insights into the data. For datasets with more than one input dimension, it is straight-forward to build GP models as a sum of one-dimensional functions, or known pairs (or triplets, etc.) of interacting inputs. In the statistics literature, Generalized Additive Models (GAMs) (Hastie & Tibshirani, 2017; Wood, 2017), are often built using sums of splines over either each input independently or over carefully selected sets of inputs.

[1]Amazon, Cambridge, United Kingdom. Correspondence to: Xiaoyu Lu <luxiaoyu@amazon.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

From the standpoint of explainable and interpretable machine learning, additive Gaussian processes such as those considered within Kaufman & Sain (2010); Duvenaud et al. (2011); Timonen et al. (2021) offer the promise of automatically discovering relevant features and combinations of features, through learning of a kernel with parameterized additive structure. In particular, Duvenaud et al. (2011) proposed a kernel which allows additive interactions of all orders, ranging from first order terms to the interactions between all the features. An efficient computation scheme was proposed for avoiding the exponentially large sum required over combinations.

In this work, we build on and challenge the findings of Duvenaud et al. (2011), where the experimental results suggest that high order terms are required to model some of the regression and classification datasets. We show that the dimensionality of the models constructed is considerably higher than necessary: for example, their model of the 8-dimensional *pumadyn* dataset requires an 8-dimensional interaction whereas our proposed model requires only 2-dimensional interactions (see Figure 1). A full comparison on all the datasets used in Duvenaud et al. (2011) is provided in Section 5.1: in all cases, we find that a small number of low-dimensional terms are needed to achieve similar or better performance. We posit that the high dimensional nature of their models are due to two issues: an identifiability issue with the summed components; and the way that the contribution of a component to the overall model is measured.

We solve the identifiability issue by borrowing an idea from Durrande et al. (2012), where the components of the additive model are orthogonalized. We call the resulting kernel *orthogonal additive kernel* (OAK), which can produce highly parsimonious models of the datasets studied in Duvenaud et al. (2011), as well as more recent larger datasets. Plumlee & Joseph (2018) tackles a slightly different identifiability issue by proposing a GP whose stochastic part is orthogonal to the mean part. We measure the contribution of any component to the overall model using a Sobol index (Sobol, 1993; Owen, 2014) which is shown to be analytic for the OAK model. We see in Section 5.1 that the pumadyn dataset can be modelled using a sum of only three components – one two-dimensional interaction function and two one-dimensional functions. These parsimo-

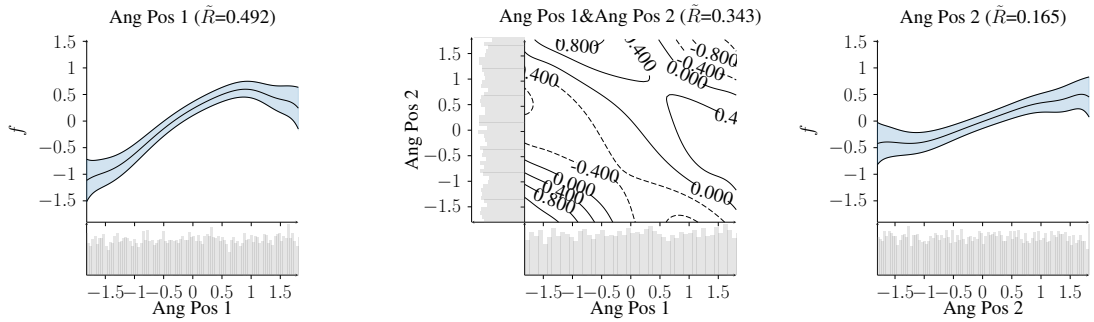


Figure 1. Visualization of the decomposed functions with highest Sobol indices for the pumadyn dataset. On the horizontal axis we plot different feature x_i, on the vertical aixs is its corresponding function f_i. For two-way interaction terms, a contour plot is used. Grey bars represent histograms of input features, black solid lines represent posterior mean GP, blue shaded area represents ± 2 standard deviations confidence interval from the GP model. $\tilde{R}$ in the brackets represent (normalized) Sobol indices. We can observe that over 99% of the variance can be explained with only these three terms: two first order terms and one interaction term between them. We reach optimal model performance with only these three terms (Figure 5).

nious models are highly explainable since each effect of a component can be examined in a simple plot, yet the model remains powerful: the predictive performance is on par with or better than either the original additive model or a full squared exponential GP model. In a case study on the SUSY physics dataset (Section 5.2), our method produces a low dimensional model with only ten one-dimensional and two-dimensional terms that outperforms the dropout-based neural network baseline. On another case study of a contemporary dataset of customer churn (Section 5.3), our method outperforms the XGBoost baseline whilst providing low-dimensional components that offer insights into business problems.

Finally, since the OAK method is using a new kernel within a standard GP formulation, we are able to scale the method using sparse GP methods. We show in Section 3.5 that the scalability of a sparse GP with the OAK kernel is favorable to that of a squared exponential kernel, since the eigenspectrum of our low dimensional model is more easily represented by an inducing point formulation. We build on recent work (Burt et al., 2019) to show increased convergence rates for sparse GPs with our proposed kernel.

Our main contribution is to combine the orthogonality constraint in Durrande et al. (2012) with the additive model in Duvenaud et al. (2011) that utilizes the Newton-Girard trick, where computationally complexity scales polynomially rather than exponentially with the number of features. We draw the link to functional ANOVA (FANOVA) decomposition (Owen, 2014; Chastaing & Le Gratiet, 2015; Ginsbourger et al., 2016) and quantify the contribution of each component with analytic Sobol indices. We have conducted extensive sets of regression and classification experiments to show its practical value. The resulting model is parsimonious and interpretable, requiring minimal model tuning.

The remainder of this manuscript is organized as follows. In Section 2 we recap the additive model used in Duvenaud et al. (2011) and propose OAK in Section 3. We introduce Sobol index in Section 4 and discuss its relationship with functional ANOVA decomposition and OAK. Experimental results are given in Section 5 and we conclude in Section 6. Our code is available at `https://github.com/amzn/orthogonal-additive-gaussian-processes`.

2. Background and Related Work

We are interested in modeling output y as a function of D-dimensional input features $\mathbf{x} := (x_1, \cdots x_D)$ with a hidden function $f(\mathbf{x})$. Duvenaud et al. (2011) considers building a GP model with the additive structure:

$$f(\mathbf{x}) = f_1(x_1) + f_2(x_2) + \cdots + f_{12}(x_1, x_2) + \cdots + f_{12\ldots D}(x_1, x_2, \cdots x_D). \quad (1)$$

In a GP model, the additive structure of the function decomposition is enforced through the structure of the kernel, whose decomposition can be constructed as follows: first assign each dimension $i \in \{1 \ldots D\}$ a one-dimensional *base kernel* $k_i(x_i, x_i')$; then define the first order, second order and d^{th} order additive kernel as:

$$k_{add_1}(x, x') = \sigma_1^2 \sum_{i=1}^{D} k_i(x_i, x_i'),$$
$$k_{add_2}(x, x') = \sigma_2^2 \sum_{i=1}^{D} \sum_{j=i+1}^{D} k_i(x_i, x_i') k_j(x_j, x_j'), \quad (2)$$
$$k_{add_d}(x, x') = \sigma_d^2 \sum_{1 \le i_1 \le i_2 \le \cdots \le i_d \le D} \left[\prod_{l=1}^{d} k_{i_l}(x_{i_l}, x_{i_l}') \right],$$

with the kernel then constructed by summing over all of the orders up to the dimensionality of the data. The parameters σ_d^2 control the relative importance of high-dimensional and

low-dimensional functions in the sum: we shall see later in this work that the high-order terms can be set to zero for all the datasets we consider using our proposed method, effectively truncating the sum. Although there can be a very large number of terms in the kernel, Duvenaud et al. (2011) proposed an algorithm based on the Newton-Girard identity to efficiently compute the kernel in polynomial time, see detailed algorithm in Appendix A.

When it comes to measuring the importance of each interaction, Duvenaud et al. (2011) proposed considering the estimated parameters σ_d^2. In Section 3 we show through a simple example that these parameters are unidentifiable. We follow a different approach using Sobol indices (e.g. Sobol, 1993; Muehlenstaedt et al., 2012; Owen, 2014) to weigh the importance of different components of the construction.

Imposing an orthogonal constraint on additive kernel components was proposed by Durrande et al. (2012) and extended in Durrande et al. (2013) and Märtens (2019). Denoting the constrained kernel by $\tilde{k}$, the kernel was constructed in the form $k(\mathbf{x}, \mathbf{x}') = \prod_{d=1}^{D}(1 + \tilde{k}_d(x_d, x_d'))$, which does not allow for control of the importance of different orders, *cf.* (2), and they did not apply the kernel in the context of GP regression, so were not able to learn kernel parameters. Märtens et al. (2019) also extended Durrande et al. (2012), building low-dimensional latent variable models where the latent and observed features are orthogonal. In the current paper, we focus on the interpretability and parsimony of the orthogonal models for regression and classification tasks in a practical setting. In particular, we extend to large numbers of features through the efficient Newton-Girard procedure of Duvenaud et al. (2011).

3. Orthogonality

With the decomposition in (1), we may learn different models that give the same predictions: this is due to the non-identifiability of the summed functions (Ginsbourger et al., 2008; Märtens, 2019). Assume a two-dimensional problem:

$$f(x_1, x_2) = f_1(x_1) + f_2(x_2)\,, \tag{3}$$

with the true functional decomposition f_1 and f_2, then

$$f(x_1, x_2) = (f_1(x_1) + \Delta) + (f_2(x_2) - \Delta) \tag{4}$$

is a valid decomposition for any value of Δ. In other words, there are infinitely many possible decompositions of f. This is not desirable because it makes interpretability difficult: which of the decompositions should one choose? Moreover, higher order terms can absorb effects from lower order terms and one may learn a model that is more complicated than needed, as we will now illustrate.

Take a two-dimensional example with true decomposition:

$$f(x_1, x_2) = x_1^2 - 2x_2 + \cos(3x_1)\sin(5x_2)\,. \tag{5}$$

We sample x_1 and x_2 uniformly on $(-1, 1)$ and generate $y \sim f(x_1, x_2) + \epsilon$ with f in (5) and $\epsilon \sim \mathcal{N}(0, 0.01)$. We then fit an additive GP model (Duvenaud et al., 2011) with squared exponential base kernels. We learn the kernel parameters and likelihood (noise) variance using maximum likelihood. The experiment is repeated with 9 random seeds and three unique local optima (i.e., 3 sets of hyperparameters) are discovered. We show posterior functions for one of the local optima in Figure 2 (top) (details in Appendix I).

In Figure 2 (top) we observe that the functions f_1 and f_2 have large (marginal) variance, meaning the model is less certain in isolating individual effects from other terms. In Figure 2d, we plot the interaction term with respect to x_1 by taking the average of $f_{12}(x_1, x_2)$ over x_2, i.e., $\mathbb{E}_{X_2}[f_{12}(x_1, x_2)]$. Figure 2e is a similar plot of x_2 by marginalizing out x_1. The quadratic shape in Figure 2d and the linear trend in Figure 2e show that the interaction term is capturing the individual effect of f_1 and f_2. In other words, higher order terms absorb the effect of lower order terms. The reverse can also be true, see Appendix I.

3.1. GP with Orthogonal Additive Kernel

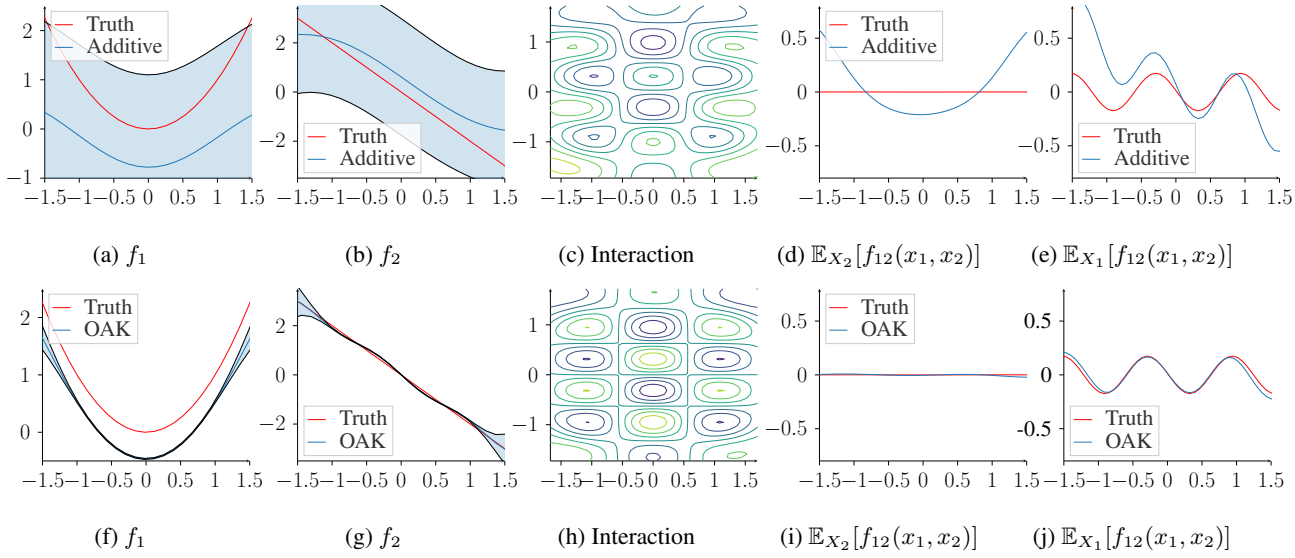
To mitigate the identifiability problem, we incorporate an idea from Durrande et al. (2012), where a constraint is used on each base kernel such that the integral of each function $\{f_i\}_{i=1}^D$ with respect to the input measure is zero. For f with non-zero mean, the offset can be modelled using a constant kernel, resulting in a unique decomposition. Our model takes the same form as (1), except adding an additional GP f_0 with constant kernel. Define $[D] := \{1, \cdots, D\}$, we constrain each f_i to satisfy the *orthogonality constraint*:

$$\int_{\mathcal{X}_i} f_i(x_i)p_i(x_i)dx_i = 0, \tag{6}$$

for $i \in [D]$, where $\mathcal{X}_i$ and p_i are the sample space and the density for input feature x_i respectively.

We now describe how we can construct the kernel for each f_i. For each feature i with base kernel k_i, it can be shown that conditioning on $S_i := \int f_i(x_i)p_i(x_i)dx_i = 0$, the process f is another GP with a modified kernel $\tilde{k}_i$:

$$f_i(\cdot)\,\Big|\,\int f_i(x_i)p_i(x_i)dx_i = 0 \sim \mathcal{GP}(0, \tilde{k}_i), \tag{7}$$



(a) f_1 (b) f_2 (c) Interaction (d) $\mathbb{E}_{X_2}[f_{12}(x_1, x_2)]$ (e) $\mathbb{E}_{X_1}[f_{12}(x_1, x_2)]$

(f) f_1 (g) f_2 (h) Interaction (i) $\mathbb{E}_{X_2}[f_{12}(x_1, x_2)]$ (j) $\mathbb{E}_{X_1}[f_{12}(x_1, x_2)]$

Figure 2. Illustration of the non-identifiability of the additive GP model in Duvenaud et al. (2011) on the two-dimensional problem. Top row: additive GP model; bottom row: OAK model. Red and blue lines represent the true and learned posterior mean functions respectively, blue shaded area represent ± 2 standard deviation. From left to right: posterior of f_1 and f_2; posterior mean of f_{12}; marginal plot for f_1 in the interaction term ($\mathbb{E}_{X_2}[f_{12}(x_1, x_2)]$); marginal plot for f_2 in the interaction term ($\mathbb{E}_{X_1}[f_{12}(x_1, x_2)]$). Note how the quadratic shape in Figure 2a and the linear trend in Figure 2b are captured in the higher order terms (Figure 2d and 2e). OAK correctly identifies the true additive components with smaller uncertainties. Note that the constant gap between the truth and OAK in Figure 2f is expected and is captured with the constant kernel.

where

$$\begin{aligned}\tilde{k}_i(x_i, x_i') =& k_i(x_i, x_i') - \mathbb{E}[S_i f_i(x_i)]\mathbb{E}[S_i^2]^{-1}\mathbb{E}[S_i f_i(x_i')]\,,\\ \mathbb{E}[S_i f_i(\cdot)] =& \int p_i(x_i) k_i(x_i, \cdot) dx_i\,,\\ \mathbb{E}[S_i^2] =& \int\int p_i(x_i) p_i(x_i') k_i(x_i, x_i') dx_i dx_i'\,. \end{aligned} \tag{8}$$

We call $\tilde{k}_i$ the *constrained* kernel. For higher order interaction terms, we desire the constraint $\int_{\mathcal{X}_i} f_u(\mathbf{x}_u) p_i(x_i) dx_i = 0\ \forall i \in u$ where $\mathbf{x}_u := \{x_i\}_{i \in u}$. This is achieved by simply taking the product of one-dimensional constrained kernels: for any $u \subseteq [D]$,

$$\tilde{k}_u(x, x') = \prod_{i \in u} \tilde{k}_i(x_i, x_i'). \tag{9}$$

A function f_u drawn from a GP with the constrained kernel $\tilde{k}_u$ satisfies the orthogonality condition assuming independent input features, see proof in Appendix B.

Since the orthogonal construction can be achieved by using sums and products of kernels, we can construct our model by plugging in the constrained kernel (8) to the sum structure (2). We call this the *Orthogonal Additive Kernel* (OAK). Note that under the orthogonality constraint, the decomposition in (2) is identifiable since it is precisely the FANOVA decomposition, see details in Section 4.

3.2. Base Kernel

We choose to use a squared exponential kernel for continuous features as the base kernel due to its analytic solution with orthogonality constraints. Other kernel choices such as the Matérn kernel also leads to analytic expressions for the constrained kernel.

Specifically, for squared exponential base kernel k_i with unit variance and lengthscale l_i: $k_i(x_i, x_i') = \exp\left(-\frac{(x_i - x_i')^2}{2l_i^2}\right)$, $\tilde{k}_i$ is analytic and has a closed form solution when the input density p_i is Gaussian, mixture of Gaussian, uniform, categorical, or approximated with the empirical distribution. We hereby give results in the case of Gaussian measure: without loss of generality, assuming one-dimensional x with $p(x) = \mathcal{N}(\mu, \delta^2)$ where we drop subscript i for simplicity, the constrained squared exponential $\tilde{k}$ is:

$$\begin{aligned}\tilde{k}(x, x') :=& \exp\left(-\frac{(x - x')^2}{2l^2}\right) - \frac{l\sqrt{l^2 + 2\delta^2}}{l^2 + \delta^2} \times\\ & \exp\left(-\frac{((x - \mu)^2 + (x' - \mu)^2)}{2(l^2 + \delta^2)}\right). \end{aligned} \tag{10}$$

For other forms of input densities, please refer to Appendix D. For categorical features, we can use the categorical kernel and an empirical input density p (see Appendix C and E).

3.3. Normalizing Flow

To satisfy the Gaussian input density assumption, we use a normalizing flow (Rezende & Mohamed, 2015) to transform continuous input features to have an approximate Gaussian density. This is achieved by applying a sequence of bijective transformations on each feature, whose parameters are learnt by minimizing the KL divergence between a standard Gaussian distribution and the transformed input data. The parameters are then fixed *before* fitting the OAK model on the transformed data with approximate Gaussian densities. For details and ablation studies, see Appendix F and J.5.

3.4. Illustration

We use the example from (5) to illustrate the constrained model described above with results given in Figure 2 (bottom). We have found that the GP model with the constrained squared exponential kernel is more stable as all 9 runs using different initial configurations converge to the same hyperparameters as opposed to the unconstrained model where we have found 3 local optima. We are able to capture the correct form of first order terms and the interaction component, resulting in a better fit and better calibrated uncertainty. Note that the constant gap (vertical shift) in Figure 2f is expected since we constrained each function to have zero mean with respect to the input density, and a separate constant kernel is used to capture the gap due to the non-zero mean of f.

3.5. Sparse GP with Inducing Points

When the number of data points N is big, GP inference costs $\mathcal{O}(N^3)$ in computation which is expensive. Variational inference with sparse GP can be used to reduce the computational costs to $\mathcal{O}(NM^2)$ where M is the number of inducing variables (Titsias, 2009).

Burt et al. (2019) showed that the number of inducing points M needed for sparse GP regression with normally distributed inputs in D-dimensional space with the squared exponential kernel is $M = \mathcal{O}(log^D N)$.

In practice, one can limit the maximum order of interactions to be $\tilde{D} \leq D$. For our additive model, the number of kernels to be added is therefore $\sum_{k=1}^{\tilde{D}} \binom{D}{k}$ and the number of inducing points needed is

$$\sum_{k=1}^{\tilde{D}} \binom{D}{k} \mathcal{O}(log^k N) = \mathcal{O}\left(\binom{D}{\tilde{D}} log^{\tilde{D}} N\right).$$

The number of inducing points needed for OAK is smaller than that for the non-orthogonal case. We also verify this empirically on the pumadyn dataset with a 4:1 training-test split. We compare our model with its non-orthogonal counterpart as in (1) and a sparse GP model with squared exponential kernel. Results are displayed in Figure 3, where OAK converges much faster and needs a smaller number of inducing points to reach same/better test RMSE (additional experiments can be found in Appendix J.7).

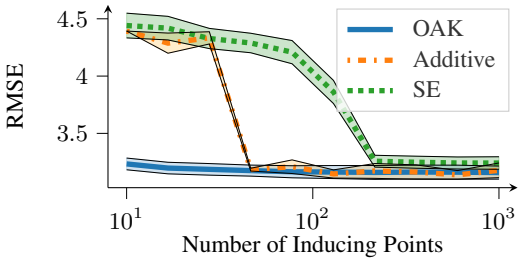


Figure 3. Test RMSE versus number of inducing points for the pumadyn dataset. Results are averaged over 5 repetitions, shaded area represents ± 1 standard deviation.

4. ANOVA Decomposition and Sobol Indices

Practitioners are often interested in the importance of features in predicting the output. For example, f may be explained using only a small number of features or interactions despite there being a large number of features. Global sensitivity analysis (Saltelli et al., 2008) is a measure of importance of input features, based on an analysis of variance (ANOVA) decomposition. Sobol indices (Sobol', 1990) are one such measure for attributing value of an output to individual features. We will see later that the Sobol indices are analytic for the OAK model.

Functional ANOVA (FANOVA) (Hoeffding & Robbins, 1948; Stone, 1994; Huang, 1998) decomposes a function $f(\mathbf{x})$ into the form $f(\mathbf{x}) = \sum_{u \subseteq [D]} f_u(\mathbf{x}_u)$, where f_u only depends on x_j for $j \in u$ and is defined recursively by

$$f_u(\mathbf{x}) = \int_{\mathcal{X}_{-u}} \left(f(\mathbf{x}) - \sum_{v \subset u} f_v(\mathbf{x}_v) \right) dP(\mathbf{x}_{-u}), \quad (11)$$

where $f_\emptyset(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$, $\mathbf{x}_{-u}$ denotes $\mathbf{x}$ excluding x_u and $P(\mathbf{x})$ denotes the distribution of $\mathbf{x}$. Applying the FANOVA decomposition to our OAK construction in (1) and (8) reveals that the functions considered in OAK are precisely the components of the FANOVA decomposition, see proof in Appendix G.3. The FANOVA decomposition associates each component with a variance. This variance is due to disturbances on the input to the function: we denote it by $\mathbb{V}_x[f_u]$. The orthogonality of OAK leads to the ANOVA identity (Owen, 2014):

$$R := \mathbb{V}_x[f(\mathbf{x})] = \sum_{u \subseteq [D]} R_u, \quad (12)$$

where $R_u := \mathbb{V}_x[f_u(\mathbf{x})]$ is defined as the Sobol index for feature set u. In other words, each R_u measures how much

variance is explained by feature set u, measuring the importance of the features. We normalize the Sobol indices such that they sum up to 1 and denote the normalized Sobol indices with $\tilde{R}$ in later sections. Similarly to Durrande et al. (2012), to assess the relative importance of a component of our model, we consider the Sobol index of the posterior mean function associated with that component: $\tilde{R}_u = \frac{\mathbb{V}_{\mathbf{x}}[m_u(\mathbf{x})]}{\mathbb{V}_{\mathbf{x}}[m(\mathbf{x})]}$, where m_u and m denote the posterior mean function of f_u and f respectively. In particular,

$$m_u(x) = \sigma^2_{|u|} \left(\odot_{i \in u} \tilde{k}_i(x_i, \mathbf{X}_i) \right) K(\mathbf{X}, \mathbf{X})^{-1} \mathbf{y} \quad (13)$$

where $K(\mathbf{X}, \mathbf{X})$ denotes the training input covariance across all inputs, $\mathbf{X}_i$ and y denote the i-th column of $\mathbf{X}$ and the vector of output observations, $\sigma^2_{|u|}$ is the associated variance parameter for the $|u|$-th order interaction and $\odot$ denotes element-wise multiplication. A similar formula for sparse GP can also be obtained. The Sobol index associated with the input set u is therefore

$$\mathbb{V}_x[m_u(x)] = \sigma^4_{|u|} y^\top K(\mathbf{X}, \mathbf{X})^{-1} \odot_{i \in u} \left(\int \tilde{k}_i(x_i, \mathbf{X}_i) \tilde{k}_i(x_i, \mathbf{X}_i)^\top \mathrm{d}p_i(x_i) \right) K(\mathbf{X}, \mathbf{X})^{-1} \mathbf{y}, \quad (14)$$

since $\mathbb{E}_x[m_u(x)] = 0$ due to the orthogonality constraint. In case of 1) constrained squared exponential kernel and a Gaussian measure or 2) binary/categorical kernel with discrete measure, the integral is tractable and can be computed analytically. More details can be found in Appendix G. Note that the Sobol index is not affected by our normalizing-flow transformation of the input, see details in Appendix G.4.

5. Experiments

Our experiment procedure runs as follows: we plug the OAK kernel in the gpflow[1] package, we then perform inference on regression problems with `gpflow.GPR` (or `gpflow.SGPR` for larger datasets); for classification tasks, we use `gpflow.SVGP` for inference We place a Gamma prior on the variance hyperparameters of the kernel, which are estimated using Maximum a Posterior (MAP). After learning the hyperparameters, we compute the Sobol index for each term including all orders of interactions up to the truncated order. Then we rank the importance of each term according to their Sobol indices and investigate how many terms are needed to give competitive model performance. Details on the procedure can be found in Appendix H.

We apply normalizing flows on all continuous features in our experiments before fitting the GP model, except for the Concrete dataset where the normalizing flow was not sufficient to transform the data and we have reverted to an empirical measure in this case. Empirically we have found that the model performance is similar with or without the normalizing flow, but the resulting model tends to be less parsimonious without the flow. More details and an ablation experimental study can be found in Appendix J.5.

We validate our model on a range of experiments, including a set of regression and classification problems from datasets used in Duvenaud et al. (2011) and additional UCI datasets, a large scale SUSY experiment and a Churn modelling problem. In our experiments we found OAK contains lower order terms without loss in predictive accuracy in contrast to Duvenaud et al. (2011) which finds higher order effects across a range of regression and classification problems. With OAK, only a small number of terms are needed in the model despite the large number of features available.

5.1. Baseline Experiments

We first duplicate the experiments in Duvenaud et al. (2011) where the number of instances and dimensionality of each dataset can be found in Appendix J.1. We use five-fold cross-validation splits and compute test RMSE for regression and area-under-the-curve (AuC) errors for classification datasets. We use a GP with a squared exponential kernel as a baseline model to compare the performance of OAK and the unconstrained additive GP model used in Duvenaud et al. (2011). For regression datasets, we set $\tilde{D} = D$; for classification problems, we set $\tilde{D} = 4$ except the Sonar dataset with $\tilde{D} = 2$ for computational considerations. We found no significant differences in performance between different models, see more details in Appendix J.

Often one is interested in understanding how much each feature or interaction of features contribute in predicting the output. For example, one may ask how much does the 3rd order interaction term affect the response, or whether some feature is more important than others in explaining the response. We first plot the cumulative Sobol index for each order of interactions in Figure 4, which is defined as the sum of Sobol indices for all terms in the same order. The results indicate that most datasets only need low order (< 3) interaction terms.

Importantly, despite there being a large number of terms including all orders of interactions terms, we found that only a few terms are needed in the model to reach competitive performance. In Figure 5 we plot model performance and cumulative Sobol as a function of the number

[1] https://github.com/GPflow/GPflow

[2] For all the classification datasets, the cumulative (normalized) Sobol indices for first order terms are found to be close to 1.

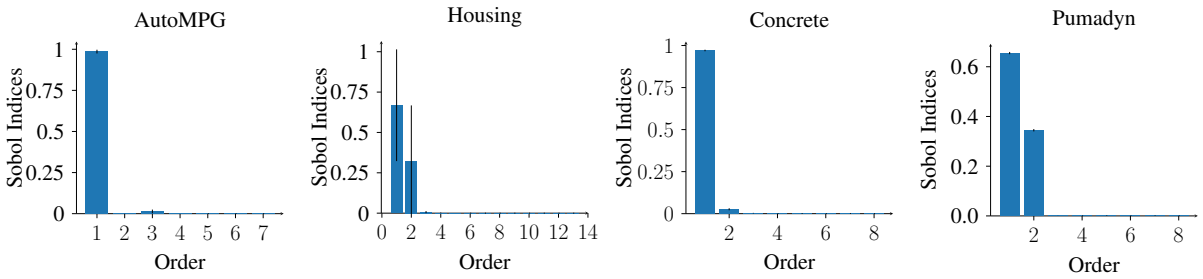


Figure 4. Sum of normalized Sobol indices for each interaction order for UCI regression problems, where bars represent one standard deviation across 5 cross-validation splits. All of the datasets require ≤ 3 order of interactions to explain the variance of the response, indicating the OAK model is able to find low dimensional representation[3].

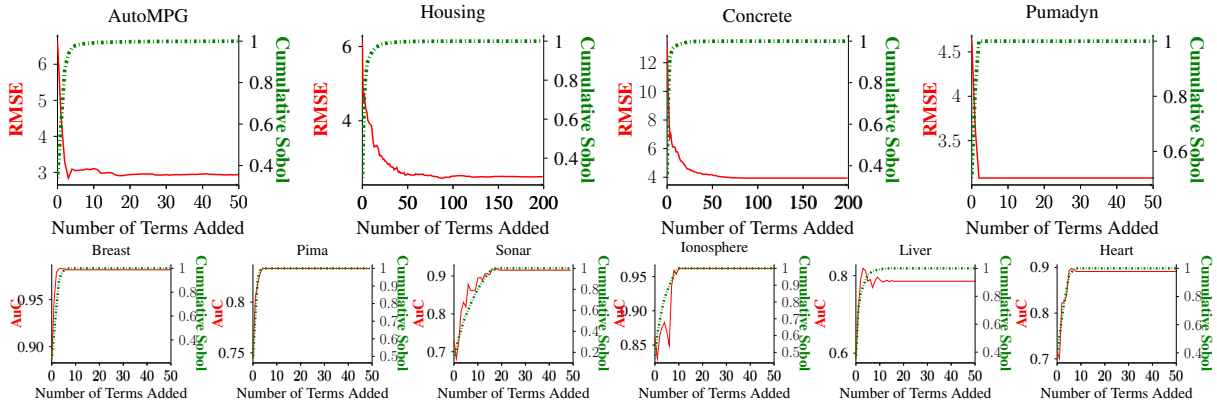


Figure 5. Model performance and cumulative Sobol index versus number of terms added ranked by the Sobol index. We use test RMSE and area-under-the-curve (AuC) as the evaluation metric for regression problems (top) and classification problems (bottom) respectively. Red solid lines represent test RMSE (top) and test AuC (bottom), green dashed lines represent cumulative (normalized) Sobol index.

of terms added, where the terms to add are ranked by their Sobol indices from highest to lowest. We report test RMSE and AuC for regression and classification problems respectively[3] .

For each dataset with dimension D and truncated maximum order of interaction $\tilde{D}$, the total number of terms is $\sum_{d=1}^{\tilde{D}} \binom{D}{d}$ (127 for autoMPG and 41448 for ionosphere datasets to give a sense of the scale, details in Appendix J.2). We can observe the strong correlation between cumulative Sobol index and model performance. Only a few number of terms are needed before the model converges, indicating further terms add little value and OAK is able to find simple representations without loss of model performance. We further verify its parsimony by comparing the interaction order variance hyperparameter σ_d^2, see Appendix J.4.

In particular, unlike in Duvenaud et al. (2011) where an 8-dimensional interaction is required, we are able to reach the same model performance with only two first order terms and one second order term, which explain $> 99\%$ variance in f. Due to the advantages of low-dimensional representation, we can visualise the decomposed functions with highest Sobol indices easily (Figure 1). For completeness, we have also conducted experiments with the kernel $\prod_d(1+\tilde{k}_d)$ used in Duvenaud et al. (2011), but using the constrained $\tilde{k}_d$. We found this kernel is harder to optimize and numerically less stable; the model performance is similar but the resulting model is less parsimonious (see Appendix J.6).

We conduct further experiments on an extensive range of benchmark datasets (Salimbeni, 2018) with results displayed in Table 1. We show summary statistics including the average, median and rank across the datasets. For regression tasks we report test RMSE and log likelihood whereas for classification tasks we report test accuracy and log likelihood. We found the performance of OAK is on par or better compared with other methods. Detailed performance metrics on each dataset can be found in Appendix K.

5.2. SUSY Classification

In the next experiment we tackle a large-scale binary classification problem. The super-symmetric (SUSY[4]) dataset

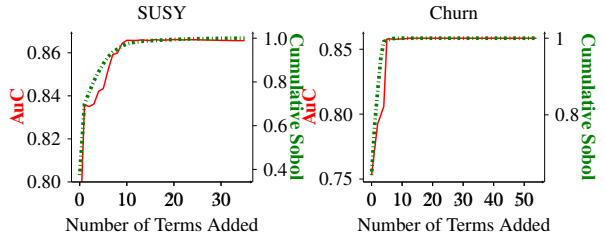
[3]We used empirical measure for Concrete dataset as its input feature distributions suggest.

[4]archive.ics.uci.edu/ml/datasets/SUSY

	Aggregation	OAK	Linear	SVGP	SVM	KNN	GBM	AdaBoost	MLP
Regression RMSE	avg	0.475	6.157	0.478	0.484	0.518	0.455	0.581	**0.445**
	median	0.376	0.736	0.397	0.419	0.454	**0.343**	0.580	0.361
	avg rank	3.583	6.625	4.083	4.208	4.958	**3.208**	5.750	3.583
Regression Log Likelihood	avg	**-0.229**	-0.946	-0.295	-0.585	-0.638	-0.652	-0.730	-0.891
	median	**-0.409**	-1.096	-0.512	-0.609	-0.738	-0.671	-0.875	-0.471
	avg rank	**5.583**	3.625	5.042	4.833	3.917	4.292	3.583	5.125
Classification Accuracy	avg	**0.872**	0.835	0.859	0.857	0.836	0.870	0.859	0.863
	median	0.898	0.832	0.864	0.850	0.863	**0.900**	0.892	0.873
	avg rank	**5.569**	4.224	4.741	4.500	2.983	5.224	4.207	4.552
Classification Log Likelihood	avg	**-0.267**	-0.338	-0.291	-0.306	-0.899	-0.283	-0.459	-0.306
	median	-0.280	-0.389	-0.307	-0.352	-1.088	**-0.256**	-0.584	-0.362
	avg rank	5.862	4.276	**5.931**	4.690	2.138	5.379	2.897	4.828

Table 1. Experimental results on additional benchmark datasets. Average results over 24 regression datasets shown in terms of test RMSE and log likelihood (top two blocks). Average results over 29 classification datasets shown in terms of accuracy and log likelihood (bottom two blocks). Higher is better except for RMSE. SVGP=Stochastic Variational GP, using GPflow (Hensman et al., 2015); SVM=Support Vector Machine, KNN=K-nearest-neighbours, GBM=Gradient Boosting Machine, MLP=Multi-layer Perceptron (all using Scikit-learn defaults). Results compiled using the Bayesian Benchmarks repo (Salimbeni, 2018). Full results are shown in Appendix K.

contains 5 million instances with 8 low level kinematic properties, where the task is to predict whether a signal process produces super-symmetric particles or not. We truncate $\tilde{D} = 2$ for computational consideration.



(a) SUSY (b) churn Modelling

Figure 6. AuC as a function of number of terms added ranked by their Sobol indices for the SUSY (left) and Churn modelling (right) experiments. Red solid lines and green dashed lines represent test AuC and cumulative (normalized) Sobol respectively.

We use the same training-test split as in Dutordoir et al. (2020). We fit a sparse variational GP (SVGP) model with OAK and optimize the variational parameters and hyper-parameters with natural gradients and Adam respectively. Number of inducing points and mini batch size are set to be 800 and 1024 respectively.

Model performance are reported in Table 2 where the OAK model achieves similar or better performance compared with other deep learning models. Top 10 important functional components are displayed in Figure 7, which contain five first order terms and five second order terms. In particular, a signal process is more likely to produce super-symmetric particles if there is higher missing energy magnitude; higher lepton 1 pT or lower lepton 2 pT. For lepton 1 eta or lepton 2 eta, the probability first increases and then decreases with increasing values of eta. In Figure 6a we can

SUSY		Churn	
Method	AuC	Method	AuC
BDT*	0.850 ± 0.003	XGBoost	0.853 ± 0.008
NN*	0.867 ± 0.002	MLP*	0.846 ± 0.013
$\text{NN}_{\text{dropout}}$*	0.856 ± 0.001	Sparse MLP*	0.828 ± 0.007
SVGP(SE)*	0.852 ± 0.002	TabTransformer*	0.856 ± 0.005
VISH*	0.859 ± 0.001	TabNet*	0.785 ± 0.024
OAK	0.865 ± 0.0004	OAK	0.856 ± 0.009

Table 2. Performance comparison for SUSY (left) and Churn modelling (right). The mean AuC is reported with one standard deviation, with 5 repetitions (SUSY) and 5 cross-validation splits (Churn) respectively. Larger is better. Results with * are quoted from Dutordoir et al. (2020) and Huang et al. (2020).

observe that with these 10 terms, we are able to reach the optimal AuC and capture 96% of the variance in f. This further shows that the OAK model is able to reach competitive performance while having a simple, interpretable representation.

5.3. Churn Modelling

Next we look at Churn Modelling problem available from Kaggle[5]. This data set contains details of a bank's customers where the goal is to predict whether the customer leaves the bank or continues to be a customer. There are 10 features including a mix of continuous and categorical variables such as age, gender, credit score, etc.. We truncate the maximum order of interactions to be 2 for computational consideration. We compare model performance with XGBoost, MLP, TabNet and TabTransformer with the same training-test split (4:1) as in Huang et al. (2020). Test AuC are reported in Table 2. We outperform or are as accurate as all of the baseline models with increased interpretability.

In Figure 6b we observe that only $\approx$ 5 terms are needed

[5] www.kaggle.com/shrutimechlearn/churn-modelling

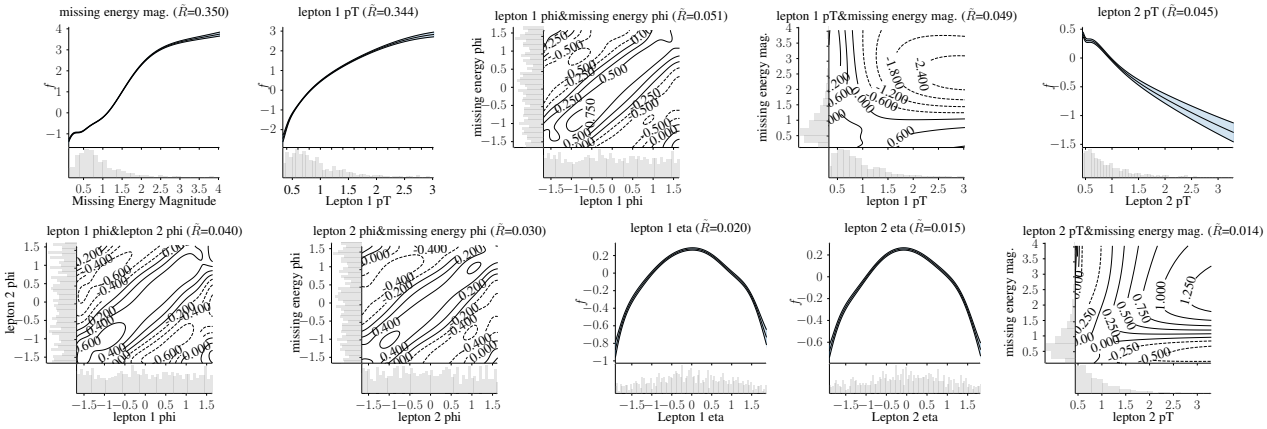


Figure 7. Decomposition of top 10 important functions for SUSY dataset, ranked by their Sobol indices. Blue shaded area represents uncertainties with two standard deviation. Grey shaded area represent histograms of input features. $\tilde{R}$ in the brackets denote (normalized) Sobol index. Missing energy magnitude and lepton 1 pT are the two most important features which explain $\approx 70\%$ of the variance in the model f, and they both have a positive impact where a signal process is more likely to produces semi-symmetric particles when missing energy magnitude and lepton 1 pT are high.

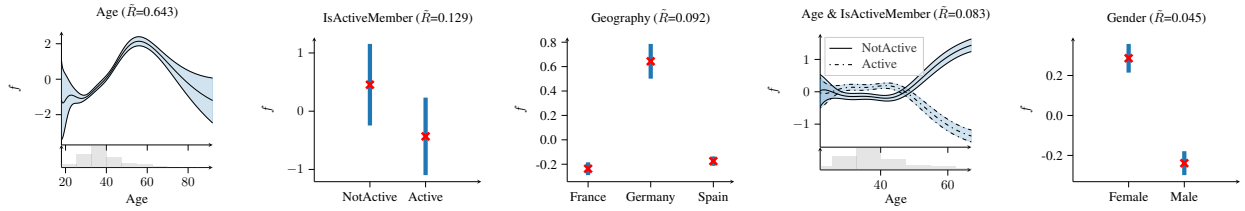


Figure 8. Decomposition of top 5 important functions for Churn dataset, ranked by their Sobol indices. Blue shaded area represents uncertainties with two standard deviation. $\tilde{R}$ in the brackets denote (normalized) Sobol index. Age is the most important feature in predicting whether a customer leaves the bank or not, typically as one gets older (but younger than 55), he/she is more likely to leave the bank. Non-active members, female customers and German customers are more likely to leave the bank compared to their counterparts. The interaction between age and whether a customer is active also contributes to the probability: older non-active customers and younger active customers are more likely to churn.

for the model to achieve optimal performance. We plot the top 5 important features/interactions in Figure 8 based on Sobol indices, which contain four first order terms and one interaction term between Age and IsActiveMember with the following insights: Age is the most important feature in predicting whether a customer leaves the bank, and generally the older a person is, more likely they will leave the bank; more active members are less likely to leave; German people are more likely to leave compared with French and Spanish; women are more likely to leave compared with men. The interaction between Age and IsActiveMember says that for less active customers, older people are more likely to leave the bank whereas for active members, older customers are more likely to stay.

6. Conclusion

In this work, we have proposed a Gaussian process model with orthogonal additive kernel (OAK) that enables inference of low-dimensional representations that are identifiable and interpretable. The resulting model has an analytic form for the Sobol indices which can be used to rank importance of features and interactions. We have shown that the OAK model allows inference of low-dimensional representations whilst achieving state-of-the-art predictive performance on a range of both regression and classification tasks. We are surprised to find out all the datasets we have experimented with can be modelled using low dimensional functions.

One limitation of our work is that we implicitly assumed independence between input features and independent, identically distributed Gaussian noise. Future work can extend our approach to non-independent input features and examine the effect of heteroscedastic noise using latent variable models. Another interesting direction of work is to extend OAK to Bayesian optimization and experimental design leveraging the inferred low-order representation.

Acknowledgement

The authors would like to thank Nicolas Durrande for insightful discussions, especially around the orthogonal-

ity construction. Thanks to George Michailidis, Dominic Richards and François-Xavier Aubet for valuable feedback on the manuscript, helpful discussions on the connections to Sobol indices, and advice on ICML rebuttals. We thank Vincent Dutordoir and Stefanos Eleftheriadis for sharing code and for help with natural gradient and Adam optimization for SVGP models. Finally, thanks to David Duvenaud, Hannes Nickisch and Carl Rasmussen whose open code and thoughtful paper inspired this work.

References

Burt, D., Rasmussen, C. E., and Van Der Wilk, M. Rates of convergence for sparse variational Gaussian process regression. In *International Conference on Machine Learning*, pp. 862–871. PMLR, 2019. (Cited on 2, 5, 24)

Chastaing, G. and Le Gratiet, L. Anova decomposition of conditional Gaussian processes for sensitivity analysis with dependent inputs. *Journal of Statistical Computation and Simulation*, 85(11):2164–2186, 2015. (Cited on 2)

Durrande, N., Ginsbourger, D., and Roustant, O. Additive covariance kernels for high-dimensional Gaussian process modeling. In *Annales de la faculté des sciences de Toulouse Mathématiques*, volume 21, pp. 481–499. Université Paul Sabatier, Toulouse, 2012. (Cited on 1, 2, 3, 6)

Durrande, N., Ginsbourger, D., Roustant, O., and Carraro, L. Anova kernels and rkhs of zero mean functions for model-based sensitivity analysis. *Journal of Multivariate Analysis*, 115:57–67, 2013. (Cited on 3)

Dutordoir, V., Durrande, N., and Hensman, J. Sparse Gaussian processes with spherical harmonic features. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2793–2802. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/dutordoir20a.html. (Cited on 8)

Duvenaud, D. K., Nickisch, H., and Rasmussen, C. Additive Gaussian processes. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/4c5bde74a8f110656874902f07378009-Paper.pdf. (Cited on 1, 2, 3, 4, 6, 7, 19, 21, 22, 23)

Ginsbourger, D., Helbert, C., and Carraro, L. Discrete mixtures of kernels for kriging-based optimization. *Quality and Reliability Engineering International*, 24(6):681–691, 2008. (Cited on 3)

Ginsbourger, D., Roustant, O., Schuhmacher, D., Durrande, N., and Lenz, N. On anova decompositions of kernels and Gaussian random field paths. In *Monte Carlo and Quasi-Monte Carlo Methods*, pp. 315–330. Springer, 2016. (Cited on 2)

Hastie, T. J. and Tibshirani, R. J. *Generalized additive models*. Routledge, 2017. (Cited on 1)

Hensman, J. A simple demonstration of coregionalization. https://gpflow.readthedocs.io/en/awav-documentation/notebooks/coreg_demo.html, 2016. (Cited on 13)

Hensman, J., Matthews, A., and Ghahramani, Z. Scalable variational Gaussian process classification. In *Artificial Intelligence and Statistics*, pp. 351–360. PMLR, 2015. (Cited on 8, 18)

Hoeffding, W. and Robbins, H. The central limit theorem for dependent random variables. *Duke Mathematical Journal*, 15(3):773–780, 1948. (Cited on 5)

Huang, J. Z. Projection estimation in multiple regression with application to functional anova models. *The annals of statistics*, 26(1):242–272, 1998. (Cited on 5)

Huang, X., Khetan, A., Cvitkovic, M., and Karnin, Z. Tabtransformer: Tabular data modeling using contextual embeddings, 2020. (Cited on 8)

Kaufman, C. G. and Sain, S. R. Bayesian functional {ANOVA} modeling using Gaussian process prior distributions. *Bayesian Analysis*, 5(1):123–149, 2010. (Cited on 1)

Märtens, K. *Enabling feature-level interpretability in nonlinear latent variable models: a synthesis of statistical and machine learning techniques*. PhD thesis, University of Oxford, 2019. (Cited on 3)

Märtens, K., Campbell, K., and Yau, C. Decomposing feature-level variation with covariate Gaussian process latent variable models. In *International Conference on Machine Learning*, pp. 4372–4381. PMLR, 2019. (Cited on 3)

Muehlenstaedt, T., Roustant, O., Carraro, L., and Kuhnt, S. Data-driven kriging models based on fanova-decomposition. *Statistics and Computing*, 22(3):723–738, 2012. (Cited on 3)

Owen, A. B. Sobol'indices and shapley value. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):245–251, 2014. (Cited on 1, 2, 3, 5)

Plumlee, M. and Joseph, V. R. Orthogonal Gaussian process models. *Statistica Sinica*, pp. 601–619, 2018. (Cited on 1)

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538. PMLR, 2015. (Cited on 5)

Salimbeni, H. Bayesian benchmarks. https://github.com/hughsalimbeni/bayesian_benchmarks, 2018. (Cited on 7, 8)

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S. *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008. (Cited on 5)

Sobol', I. M. On sensitivity estimation for nonlinear mathematical models. *Matematicheskoe modelirovanie*, 2(1): 112–118, 1990. (Cited on 5)

Sobol, I. M. Sensitivity analysis for non-linear mathematical models. *Mathematical modelling and computational experiment*, 1:407–414, 1993. (Cited on 1, 3)

Stone, C. J. The use of polynomial splines and their tensor products in multivariate function estimation. *The Annals of Statistics*, pp. 118–171, 1994. (Cited on 5)

Timonen, J., Mannerström, H., Vehtari, A., and Lähdesmäki, H. lgpr: an interpretable non-parametric method for inferring covariate effects from longitudinal data. *Bioinformatics*, 37(13):1860–1867, 2021. (Cited on 1)

Titsias, M. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial intelligence and statistics*, pp. 567–574. PMLR, 2009. (Cited on 5, 18)

Wood, S. N. *Generalized additive models: an introduction with R*. CRC press, 2017. (Cited on 1)

A. Newton-Girard Method for Computing the Interacting Kernel

Algorithm 1 Newton-Girard method for computing the interacting kernel

Input: input dimension D
Input: maximum interaction order $\tilde{D}$
Input: base kernels $k_d(\cdot,\cdot), d = 1 \dots D$
Input: order variances $\sigma_l, l = 0 \dots \tilde{D}$
Data: input data $\mathbf{X}$
Output: kernel matrix $\mathbf{K}$
for $d = 1 \dots D$ **do**
  $\mathbf{K}_d[i,j] = k_d(x_{i,d}, x_{j,d})$
end for
for $\ell = 0 \dots \tilde{D}$ **do**
  $\mathbf{S}_\ell = \sum_{d=1}^{D} \mathbf{K}_d^\ell$
end for
$\mathbf{E}_0 = \mathbf{1}^{[N,N]}$
for ℓ=1$\dots \tilde{D}$ **do**
  $\mathbf{E}_\ell = \frac{1}{\ell} \sum_{k=1}^{\ell} (-1)^{k-1} \mathbf{E}_{\ell-k} \odot \mathbf{S}_k$
end for
$\mathbf{K} = \sum_{\ell=0}^{\tilde{D}} \sigma_\ell \times \mathbf{E}_\ell$

B. Orthogonality in Higher Dimension

For higher order terms, recall OAK uses the product of constrained kernels (equation (9)):

$$\tilde{k}_u(x, x') = \prod_{i \in u} \tilde{k}_i(x_i, x_i'). \tag{15}$$

We show the product of constrained kernel satisfies the orthogonality constraint in higher dimensions, i.e.,$\forall i \in u$,

$$\int_{\mathcal{X}_i} f_u(\mathbf{x}_u) p_i(x_i) dx_i = 0 \tag{16}$$

where each functional component f_u has kernel k_u.

Proof. By construction, for each function i with constrained kernel $\tilde{k}_i$, f_i satisfies the orthogonality constraint $S_i := \int_{\mathcal{X}_i} f_i(x_i) p_i(x_i) dx_i = 0$ (equation (6)), which implies that:

$$\mathbb{E}_{f_i}[S_i] = 0, \quad \mathbb{V}_{f_i}[S_i] = \int_{\mathcal{X}_i} \tilde{k}_i(\mathbf{x}_i, \mathbf{x}_i) p_i(x_i) dx_i = 0. \tag{17}$$

To prove $\int_{\mathcal{X}_i} f_u(\mathbf{x}_u) p_i(x_i) dx_i = 0$, it is sufficient to prove the mean and variance of $\int_{\mathcal{X}_i} f_u(\mathbf{x}_u) p_i(x_i) dx_i$ with respect to f_u is zero. Since we assume f_u has zero mean, the mean $\mathbb{E}_{f_u}\left[\int_{\mathcal{X}_i} f_u(\mathbf{x}_u) p_i(x_i) dx_i\right] = 0$. The variance

$$\begin{aligned}\mathbb{V}_{f_u}\left[\int_{\mathcal{X}_i} f_u(\mathbf{x}_u)p_i(x_i)dx_i\right] &= \int_{\mathcal{X}_i} \mathbb{E}_{f_u}[f_u(\mathbf{x}_u)^2]p_i(x_i)dx_i \\ &= \int_{\mathcal{X}_i} k_u(\mathbf{x}_u,\mathbf{x}_u)p_i(x_i)dx_i \\ &= \prod_{j\neq i} k_j(x_j,x_j)\int_{\mathcal{X}_i} k_i(\mathbf{x}_i,\mathbf{x}_i)p_i(x_i)dx_i = 0.\end{aligned} \tag{18}$$

□

C. Constrained Categorical Kernel

For categorical input features, we can model f with the categorical kernel as in Hensman (2016), which is constructed by a positive definite matrix A such that the categorical kernel $k(i,j) = A_{ij}$ where

$$A = WW^\top + \text{Diag}(\kappa). \tag{19}$$

The orthogonality constraint we put on f is $\int f(x)p(x)dx = 0$. Let $\mathbf{w}$ be the vector of probability measure of the input feature, i.e., $\mathbb{P}(x=i) = w_i$ for $i = 1,\cdots,M$. Define

$$B := A - \frac{A\mathbf{w}(A\mathbf{w})^\top}{\mathbf{w}^\top A\mathbf{w}}, \tag{20}$$

we claim the kernel with $\tilde{k}(i,j) = B_{ij}$ is the constrained categorical kernel. To see this, it is enough to show $\mathbb{E}_{p(i,j)}[k(i,j)] = 0$ as shown in (18):

$$\mathbb{E}_{p(i,j)}[\tilde{k}(i,j)] = \sum_{i=0}^{M}\sum_{j=0}^{M}\tilde{k}(i,j)w_iw_j = \mathbf{w}^\top A\mathbf{w} - \mathbf{w}^\top\left(\frac{A\mathbf{w}\mathbf{w}^\top A}{\mathbf{w}^\top A\mathbf{w}}\right)\mathbf{w} = 0. \tag{21}$$

D. Constrained Squared Exponential Kernel

D.1. Gaussian Measure

We prove the constrained squared exponential kernel takes the form in (10) when the input feature has Gaussian density. Assume squared exponential kernel with lengthscale l and variance σ^2, and Gaussian measure $p(x) \sim \mathcal{N}(\mu,\delta^2)$. Denote $S := \int f(x)p(x)dx$, by (8) we need to calculate:

$$\mathbb{E}_f[Sf(a)] = \int \sigma^2 p(x)\exp\left(-\frac{(x-a)^2}{2l^2}\right)dx \tag{22}$$

$$= \int \frac{\sigma^2}{\sqrt{2\pi\delta^2}}\exp\left(-\frac{(x-a)^2}{2l^2}\right)\exp\left(-\frac{(x-\mu)^2}{2\delta^2}\right)dx \tag{23}$$

$$= \int \sigma^2\sqrt{2\pi l^2}\mathcal{N}(x;a,l^2)\mathcal{N}(x;\mu,\delta^2)dx \tag{24}$$

$$= \sigma^2\sqrt{\frac{l^2}{l^2+\delta^2}}\exp\left(-\frac{(a-\mu)^2}{2(l^2+\delta^2)}\right), \tag{25}$$

and

$$\mathbb{E}_f[S^2] = \int\int p(x)p(x')k(x,x')dxdx' \tag{26}$$

$$= \int \sigma^2\sqrt{\frac{l^2}{l^2+\delta^2}}\exp\left(-\frac{(x-\mu)^2}{2(l^2+\delta^2)}\right)p(x)dx \tag{27}$$

$$= \sigma^2\sqrt{\frac{l^2}{l^2+2\delta^2}} \tag{28}$$

where the last equality follows from completing the square.

D.2. Mixture of Gaussian Measure

We can extend the Gaussian density assumption to other input distributions such as mixture of Gaussians. Suppose a fixed number of clusters K:

$$p(x) = \sum_{k=1}^{K} w_k N(\mu_k, \delta_k) \tag{29}$$

where μ_k, δ_k are the mean and variance of each cluster.

From (8), two expectations need to be calculated to compute the constrained kernel, the variance $\mathbb{E}_f[S^2]$ and the covariance $\mathbb{E}_f[Sf(x)]$ can be computed as

$$\mathbb{E}_f[S^2] = \sum_{i=1}^{K}\sum_{j=1}^{K} w_i w_j l N(\mu_i|\mu_j, l^2 + \delta_i + \delta_j)(2\pi)^{1/2}, \tag{30}$$

$$\mathbb{E}_f[Sf(x)] = \sum_{k=1}^{K} l w_k N(x|\mu_k, \delta_k + l^2)(2\pi)^{1/2} \tag{31}$$

where l is the kernel lengthscale parameter and we have assumed unit kernel variance parameter for simplicity.

E. Constrained Kernel under Empirical Measure

When input densities are far from (mixture of) Gaussian distributions, or categorical kernel is not appropriate, or one wants to use other kernels, we can use the empirical measure $p(x) = \sum_{i=1}^{M} w_i \mathbb{1}_{x=x_i}$, where $\{x_i\}_{i=1}^{M}$ are the locations of the feature and $\{w_i\}_{i=1}^{M}$ are the associated weights. We can approximate (8) with

$$\mathbb{E}_f[Sf(\cdot)] \approx \sum_{i=1}^{M} w_i k(x, x_i), \quad \mathbb{E}_f[S^2] \approx \sum_{i=1}^{M}\sum_{j=1}^{M} w_i w_j k(x_i, x_j). \tag{32}$$

F. Normalizing Flow

Specifically, let $\{x^i\}_{i=1}^{N}$ be the data for feature x with unknown underlying density $p_x(x)$. We apply a sequence of K bijective functions to obtain the transformed features z:

$$z = f_K \circ f_{K-1} \circ f_1(x) := g(x). \tag{33}$$

The density of z can be calculated as:

$$p_z(z) = \frac{1}{g'(x)} p_x(x) \approx \frac{1}{N}\sum_{i=1}^{N} \frac{1}{g'(x^i)} \mathbb{1}_{x=x^i}, \tag{34}$$

where g' denotes the derivative. We would like z to be as close to standard Gaussian distributed as possible, denote $p(z)$ to be $\mathcal{N}(0,1)$, we minimize the KL-divergence:

$$KL(p_z(z)||p(z)) = \mathbb{E}_{p_z(z)}\left[\log \frac{p_z(z)}{p(z)}\right] \tag{35}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N}\left[\log \frac{p_z(z^i)}{p(z^i)}\right] \tag{36}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left((z^i)^2 - \log g'(x^i)\right) + C \tag{37}$$

where C is some constant and we approximated p_z with its empirical distribution. The parameters of g are then learnt by minimizing this KL divergence.

G. Sobol Indices

Recall the normalized Sobol index for the posterior mean of f_u for $u \in [D]$ is:

$$\tilde{R}_u = \frac{\mathbb{V}_{\mathbf{x}}[m_u(\mathbf{x})]}{\mathbb{V}_{\mathbf{x}}[m(\mathbf{x})]}. \tag{38}$$

The posterior mean GP with component u is:

$$m_u(x) = \sigma^2_{|u|} \left(\odot_{i\in u} k_i(x_i, \mathbf{X}_i)\right) K(\mathbf{X}, \mathbf{X})^{-1} y \tag{39}$$

where $K(\mathbf{X}, \mathbf{X})$ denotes the training input covariance across all inputs, y denotes the $n \times 1$ vector of output observations, $\sigma^2_{|u|}$ is the associated variance parameter for $|u|$-th order interaction and $\odot$ denotes element-wise multiplication. A similar formula for sparse GP can also be obtained. The posterior variance with respect to the input is therefore

$$\begin{aligned}
\mathbb{V}_x[m_u(x)] &= \mathbb{V}_x\left[\sigma^2_{|u|} \left(\odot_{i\in u} k_i(x_i, \mathbf{X}_i)\right) K(\mathbf{X}, \mathbf{X})^{-1} y\right] \\
&= \sigma^4_{|u|} y^\top K(\mathbf{X}, \mathbf{X})^{-1} \mathrm{cov}\left[\odot_{i\in u} k_i(x_i, \mathbf{X}_i)\right] K(\mathbf{X}, \mathbf{X})^{-1} y \\
&= \sigma^4_{|u|} y^\top K(\mathbf{X}, \mathbf{X})^{-1} \odot_{i\in u} \left(\int k_i(x_i, \mathbf{X}_i) k_i(x_i, \mathbf{X}_i)^\top dp_i(x_i)\right) K(\mathbf{X}, \mathbf{X})^{-1} y.
\end{aligned} \tag{40}$$

In case of 1) constrained squared exponential kernel and a Gaussian measure or 2) binary/categorical kernel with discrete measure, the integral is tractable and can be computed analytically.

G.1. Sobol Index for Constrained Squared Exponential Kernel

To compute the Sobol index, we need to compute the integral in (40). Dropping subscript i for simplicity, assume one-dimensional feature X, squared exponential base kernel with lengthscale l and variance σ^2: $k(x, x') = \sigma^2 \exp\left(-\frac{1}{2l^2}(x - x')^2\right)$ and Gaussian input density $p(x) = \mathcal{N}(\mu, \delta^2)$, recall the constrained squared exponential kernel $\tilde{k}$ is :

$$\begin{aligned}
\tilde{k}(x, x') &:= k(x, x') - \frac{\sigma^2 l \sqrt{l^2 + 2\delta^2}}{l^2 + \delta^2} \exp\left(-\frac{1}{2(l^2 + \delta^2)}((x - \mu)^2 + (x' - \mu)^2)\right) && (41) \\
&:= k(x, x') - \hat{k}(x, x') && (42)
\end{aligned}$$

where

$$\hat{k}(x, x') := \frac{\sigma^2 l \sqrt{l^2 + 2\delta^2}}{l^2 + \delta^2} \exp\left(-\frac{1}{2(l^2 + \delta^2)}((x - \mu)^2 + (x' - \mu)^2)\right). \tag{43}$$

Denote $a = X_p$, $b = X_q$ respectively, The (p, q)-entry is of $\int \tilde{k}(x, X)\tilde{k}(x, X)^\top dp(x)$ in (40) is therefore

$$\begin{aligned}
\int p(x)\tilde{k}(x, a)\tilde{k}(x, b)dx &= \int p(x)k(x, a)k(x, b)dx && (44) \\
&- \int p(x)k(x, a)\hat{k}(x, b)dx && (45) \\
&- \int p(x)\hat{k}(x, a)k(x, b)dx && (46) \\
&+ \int p(x)\hat{k}(x, a)\hat{k}(x, b)dx. && (47)
\end{aligned}$$

We compute each of the term in following subsections.

G.1.1. EQUATION (44)

$$\begin{aligned}\int p(x)k(x,a)k(x,b)dx &= \int p(x)\sigma^4 \exp\left(-\frac{1}{2l^2}\left((x-a)^2+(x-b)^2\right)\right)dx \\ &= \sigma^4 \int p(x)\exp\left(-\frac{1}{2l^2}(2x^2-2(a+b)z+a^2+b^2)\right)dx \\ &= \sigma^4 \exp\left(-\frac{1}{2l^2}(x^2+y^2)\right)\int p(x)\exp\left(-\frac{1}{l^2}(x^2-(a+b)z)\right)dx \\ &= \sigma^4 \exp\left(-\frac{1}{2l^2}(a^2+b^2)\right)\exp\left(-\frac{1}{l^2}\left(\frac{a+b}{2}\right)^2\right)\int p(x)\exp\left(-\frac{1}{l^2}\left(x-\frac{a+b}{2}\right)^2\right)dx.\end{aligned}$$

Note

$$\begin{aligned}\int p(x)\exp\left(-\frac{1}{l^2}\left(x-\frac{a+b}{2}\right)^2\right)dx &= \sqrt{\pi l^2}\int \mathcal{N}(z;\mu,\delta^2)\mathcal{N}\left(x;\frac{a+b}{2},\frac{l^2}{2}\right)dx \\ &= \frac{l}{\sqrt{2\delta^2+l^2}}\exp\left(-\frac{1}{2\delta^2+l^2}\left(\mu-\frac{a+b}{2}\right)^2\right).\end{aligned}$$

Hence

$$\int p(x)k(x,a)k(x,b)dx = \frac{\sigma^4 l}{\sqrt{2\delta^2+l^2}}\exp\left(-\frac{1}{4l^2}(a-b)^2\right)\exp\left(-\frac{1}{2\delta^2+l^2}\left(\mu-\frac{a+b}{2}\right)^2\right).$$

G.1.2. EQUATION (45)

$$\int p(x)k(x,a)\hat{k}(x,b)dx = \frac{\sigma^4 l\sqrt{l^2+2\delta^2}}{l^2+\delta^2}\exp\left(-\frac{1}{2(l^2+\delta^2)}(b-\mu)^2\right)\int p(x)\exp\left(-\frac{(x-a)^2}{2l^2}-\frac{(x-\mu)^2}{2(l^2+\delta^2)}\right)dx.$$

Note

$$\begin{aligned}\int p(x)\exp\left(-\frac{(x-a)^2}{2l^2}-\frac{(x-\mu)^2}{2(l^2+\delta^2)}\right)dx &= \int p(x)\exp\left(-\frac{1}{2M^{-1}}(x-c)^2+C\right)dx \\ &= \sqrt{2\pi M^{-1}}\exp\left(-\frac{C}{2}\right)\int \mathcal{N}(x;\mu,\delta^2)\mathcal{N}(x;c,M^{-1})dx \\ &= \frac{1}{\sqrt{\delta^2 M+1}}\exp\left(-\frac{C}{2}\right)\exp\left(-\frac{1}{2(\delta^2+M^{-1})}(c-\mu)^2\right),\end{aligned}$$

where

$$M := \frac{1}{l^2}+\frac{1}{l^2+\delta^2}, \qquad c := M^{-1}\left(\frac{\mu}{l^2+\delta^2}+\frac{a}{l^2}\right) \qquad C := \frac{a^2}{l^2}+\frac{\mu^2}{l^2+\delta^2}-c^2 M.$$

Hence,

$$\int p(x)k(x,a)\hat{k}(x,b)dx = \frac{\sigma^4 l\sqrt{l^2+2\delta^2}\exp(-C/2)}{(l^2+\delta^2)\sqrt{\delta^2 M+1}}\exp\left(-\frac{1}{2(l^2+\delta^2)}(b-\mu)^2\right)\exp\left(-\frac{1}{2(\delta^2+M^{-1})}(c-\mu)^2\right).$$

G.1.3. EQUATION (46)

By symmetry, this is straight-forward by interchanging a and b in (45).

G.1.4. EQUATION (47)

$$\int p(x)\hat{k}(x,a)\hat{k}(x,b)dx = \frac{\sigma^2 l^2(l^2+2\delta^2)}{(l^2+\delta^2)^2}\exp\left(-\frac{(a-\mu)^2+(b-\mu)^2}{2(l^2+\delta^2)}\right)\int p(x)\exp\left(-\frac{(x-\mu)^2}{(l^2+\delta^2)}\right)dx.$$

Note

$$\int p(x)\exp\left(-\frac{1}{(l^2+\delta^2)}(x-\mu)^2\right)dx = \sqrt{\pi(l^2+\delta^2)}\int \mathcal{N}(x;\mu,\delta^2)\mathcal{N}\left(x;\mu,\frac{l^2+\delta^2}{2}\right)dx = \sqrt{\frac{l^2+\delta^2}{l^2+3\delta^2}}.$$

Hence

$$\int p(x)\hat{k}(x,a)\hat{k}(x,b)dx = \frac{\sigma^4 l^2(l^2+2\delta^2)\sqrt{l^2+\delta^2}}{(l^2+\delta^2)^2\sqrt{l^2+3\delta^2}}\exp\left(-\frac{1}{2(l^2+\delta^2)}((a-\mu)^2+(b-\mu)^2)\right).$$

G.2. Sobol for Empirical Measure

Assume one-dimensional feature x with empirical measure $p(x) = \sum_{i=1}^M w_i \mathbf{1}_{x=x_i}$ where M is the number of distinct empirical locations, w_i are the (normalized) empirical weights, x_i are the empirical locations for $i = 1, \cdots, M$. We can approximate the integral in (40) with

$$\int k(\mathbf{X},x)k(\mathbf{X},x)^\top dp(x) \approx \sum_{i=1}^M w_i k(\mathbf{X},x_i)k(\mathbf{X},x_i)^\top. \tag{48}$$

G.3. Proof of FANOVA for OAK

We show in this section that under the asumption that input features are independent, OAK results in the FANOVA decomposition, i.e., for each $u \subseteq [D]$, f_u with $\tilde{k}_u$ satisfies that

$$f_u(\mathbf{x}) = \int_{\mathcal{X}_{-u}} \left(f(\mathbf{x}) - \sum_{v\subset u} f_v(\mathbf{x}_v)\right) dP(\mathbf{x}_{-u}). \tag{49}$$

Proof. The right hand side writes

$$\begin{aligned}\int_{\mathcal{X}_{-u}} \left(f(\mathbf{x}) - \sum_{v\subset u} f_v(\mathbf{x}_v)\right) dP(\mathbf{x}_{-u}) &= \int_{\mathcal{X}_{-u}} \left(f_u(\mathbf{x}) + \sum_{v\not\subseteq u} f_v(\mathbf{x}_v)\right) dP(\mathbf{x}_{-u}) \\ &= f_u(\mathbf{x}_u) + \sum_{v\not\subseteq u}\int_{\mathcal{X}_{-u}} f_v(\mathbf{x}_v)dP(\mathbf{x}_{-u}).\end{aligned} \tag{50}$$

For each $v \not\subseteq u$, if $j \in [D]\setminus u$, then $j \in v$. It follows from Appendix B that

$$\int_{\mathcal{X}_j} f_v(\mathbf{x}_v)dP(\mathbf{x}_j) = \int_{\mathcal{X}_j} f_v(\mathbf{x}_v)p_j(x_j)dx_j = 0.$$

Under the assumption that input features are independent, $P(\mathbf{x}_{-u})$ factorizes and the integral in equation (50) is 0. □

G.4. Invariance of Sobol under Bijective Transformation

Let $Z \sim \mathcal{N}(0,1)$, and suppose X is a transformation of Z such that $z = g(x)$ where g is an invertible function. First note the density of X is

$$p(x) = \mathcal{N}(g(x)|0,1)\left|\frac{dg(x)}{dx}\right|. \tag{51}$$

One can rewrite a function of x as a function of z, suppose $f(x) = h(z) = h(g(x))$, the Sobol index for x can be calculated as

$$R = \int f^2(x)p(x)dx \tag{52}$$

$$= \int_{-\infty}^{\infty} h^2(g(x))p(x)dx \tag{53}$$

$$= \int_{-\infty}^{\infty} h^2(g(x))p(x)\left|\frac{dg(x)}{dx}\right|^{-1} dz \tag{54}$$

$$= \int_{-\infty}^{\infty} h^2(g(x))\mathcal{N}(g(x)|0,1)dz \tag{55}$$

$$= \int_{-\infty}^{\infty} h^2(z)\mathcal{N}(z|0,1)dz, \tag{56}$$

which is the Sobol index for Z.

H. OAK Method Summary

Choose a truncation order for the model, $\tilde{D}$.

1. For each input dimension, a kernel is assigned:
   (a) Continuous features are assigned constrained squared exponential kernels, and transformed through a normalizing flow to ensure Gaussian input density.
   (b) Discrete features are assigned a constrained binary or categorical kernel (see Appendix C).
2. Fit a Gaussian process model with OAK defined in Section 3 and the Newton Girard Trick in Algorithm 1.
   (a) For small ($N < 1000$) regression datasets, we use Exact Gaussian Process regression.
   (b) For larger regression datasets, we use Sparse GP regression (`gpflow.SGPR`, (Titsias, 2009)).
   (c) For classification datasets, we use Variational Inference (Hensman et al., 2015). For datasets with ($N > 200$), choose the number of inducing points $M = 200$, for SUSY and Churn modelling datasets, we choose $M = 800$.

   We place a Gamma prior on the variance hyperparameters of the kernel, which are estimated using MAP. The lengthscales hyperparameters are estimated by maximum likelihood, or by maximising the ELBO, appropriately.
3. Construct the Sobol index for each component and each order according to equation (14), and construct a ranking. Truncate components when the (normalized) Sobol idex is below some threshold (default 0.01).
4. Compute the posterior over the additive components identified in the above ranking, using equation (13).
5. Predict for test points by summing over the identified components.

I. Two-dimensional Toy Example

Additional experimental results for the two-dimensional example with (unconstrained) squared exponential kernel.

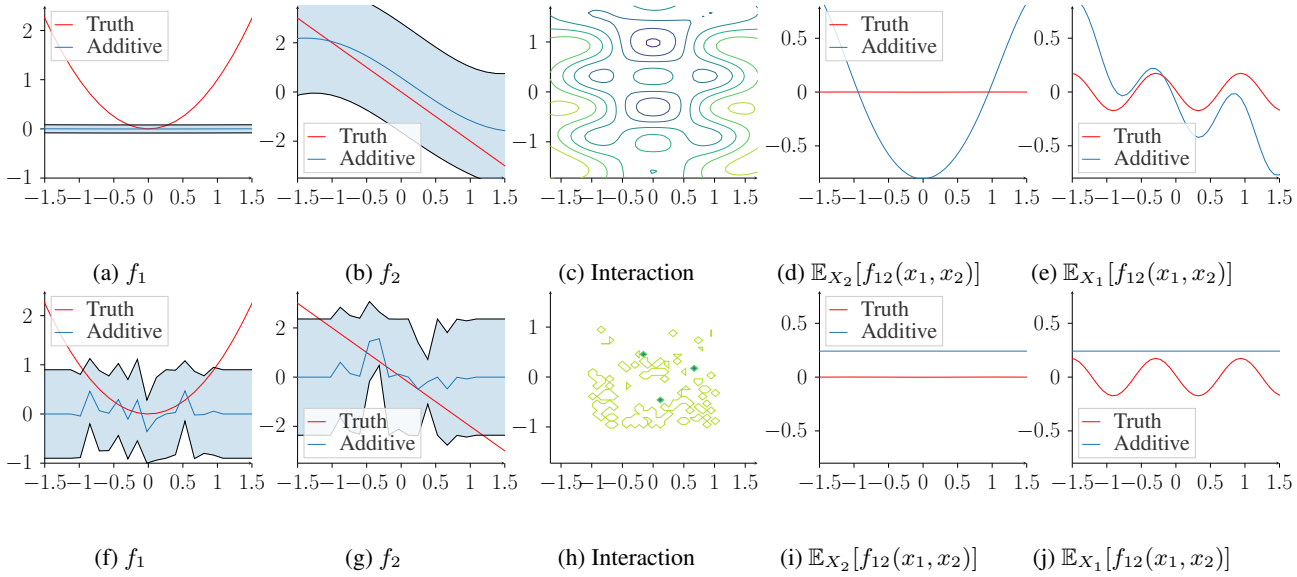


Figure 9. Two dimensional experimental results for the additive GP model in Duvenaud et al. (2011) with squared exponential kernel for the remaining two local optima. The red lines represent the true function, blue shaded area represent ± 2 standard deviation. From left to right: posterior of f_1; posterior of f_2; posterior for the interaction term; marginal plot for f_1 in the interaction term ($\mathbb{E}_{X_2}[f_{12}(x_1, x_2)]$); marginal plot for f_2 in the interaction term ($\mathbb{E}_{X_1}[f_{12}(x_1, x_2)]$). Note how the quadratic shape in Figure 9a and the linear trend in Figure 9b are captured in the higher order terms Figure 9d and Figure 9e. Vice Versa, first order terms may also absorb effect from the interaction, as Figure 9f and Figure 9g show.

J. Baseline Experimental Results

J.1. Baseline Dataset Details

Data	AutoMP	Housing	Concrete	Pumadyn	Breast	Pima	Sonar	Ionosphere	Liver	Heart
n	392	506	1030	8192	449	768	208	351	345	297
D	7	13	8	8	9	8	60	32	6	13

Table 3. Number of data and dimensionality of baseline datasets.

J.2. Total Number of Terms for Baseline Datasets

Data	AutoMP	Housing	Concrete	Pumadyn	Breast	Pima	Sonar	Ionosphere	Liver	Heart
number of terms	127	8191	255	255	255	162	1830	41448	56	1092

Table 4. Total number of additive terms in baseline datasets.

J.3. Model Performance

Model performance for baseline dataset experiments are displayed in Figure 10, where we compare percentage improvement relative to the baseline model (full GP with squared exponential kernel) for our constrained kernel and the non-constrained counterparts in Duvenaud et al. (2011). Positive values indicate superior performance compared with the baseline. Detailed performance on each of the train-split fold can be found in Figure 11, 12, 13 and 14.

J.4. Order Variance Hyperparameter Comparison

We compare the normalized variance hyperparameter $\frac{\sigma_d^2}{\sum_{d=1}^{D} \sigma_d^2}$ of each order d of interaction between OAK and Duvenaud et al. (2011), as shown in Figure 15 and 16. The results further verify that OAK model is more parsimonious and requires lower order interactions for all datasets.

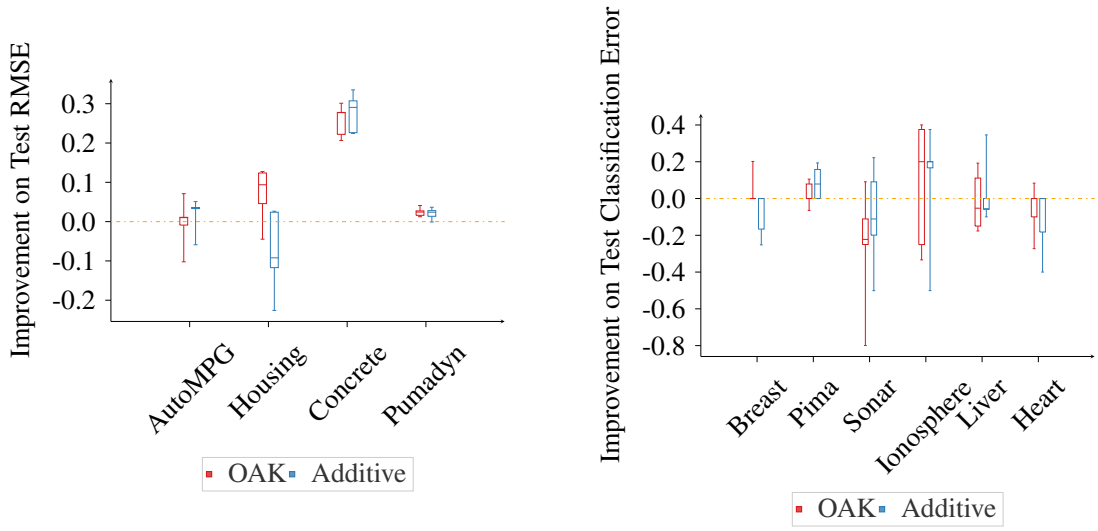


Figure 10. Test RMSE relative improvement compared with GP with squared-exponential kernel for regression (left); and test classification percentage error for classification (right). Red and blue boxes represent mean and ± 1 standard deviation over 5 train-test folds for the additive model and OAK model respectively. Horizontal axis represents different datasets; vertical axis represents model percentage improvement relative to the baseline model. Higher values are better.

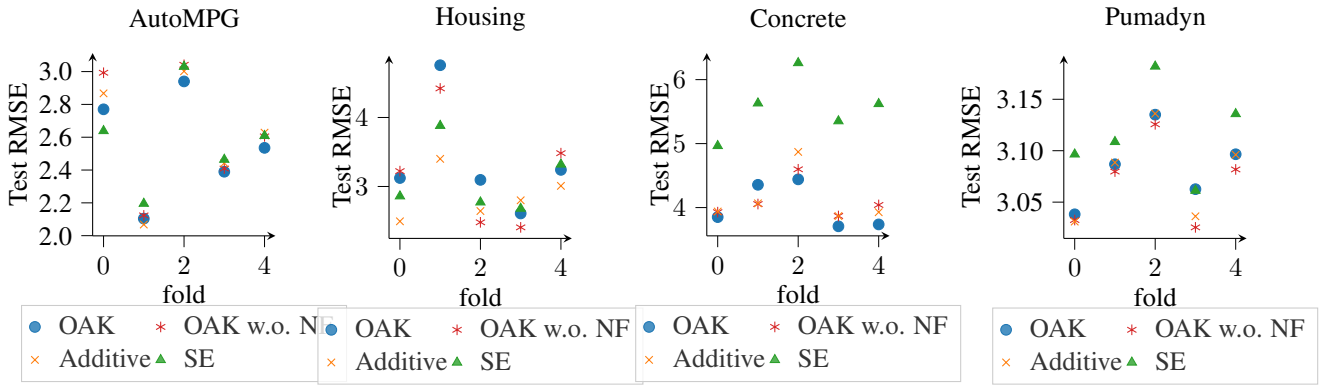


Figure 11. Test RMSE on regression datasets over 5 folds, lower is better. OAK w.o. NF stands for the OAK model without normalizing flow.

J.5. Normalizing Flow Ablation Study

Normalizing flow plays a role similar to data centering: we transform each continuous feature to be closer to Gaussian. The bijective function in the flow is a composition of shifting, scaling and sinharcsinh transformation. The parameters of the bijective functions are learned and fixed before fitting the GP model, using only the input data, not in conjunction with the hyperparametrs. For non-continuous input features we do not apply any transformation, but use the orthogonal discrete kernel described in Appendix C. We have performed an ablation study and ran experiments on all the baseline datasets where we standardize the inputs instead of using the flow. The model performance is similar (see Figure 11, 12, 13 and 14) but the resulting model tends to be less parsimonious, especially for the Housing dataset, see Figure 17 for details.

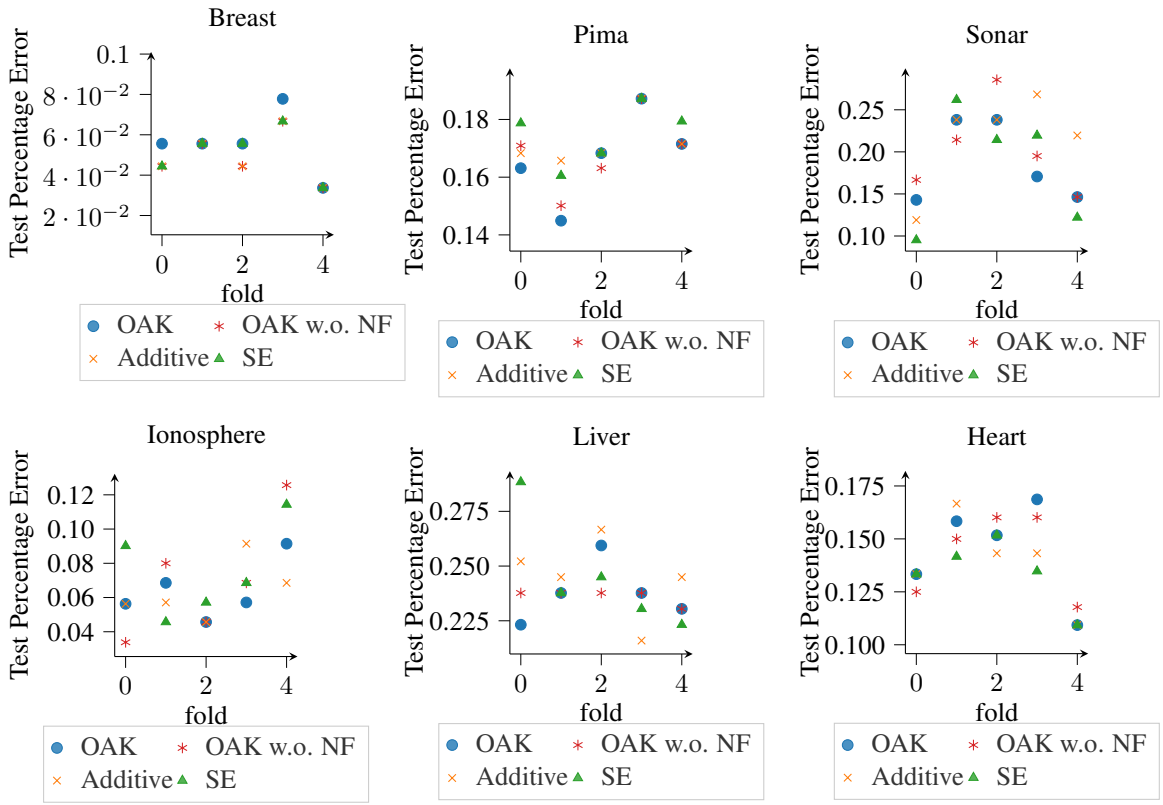


Figure 12. Test percenrage error on classification datasets over 5 folds, lower is better. OAK w.o. NF stands for the OAK model without normalizing flow.

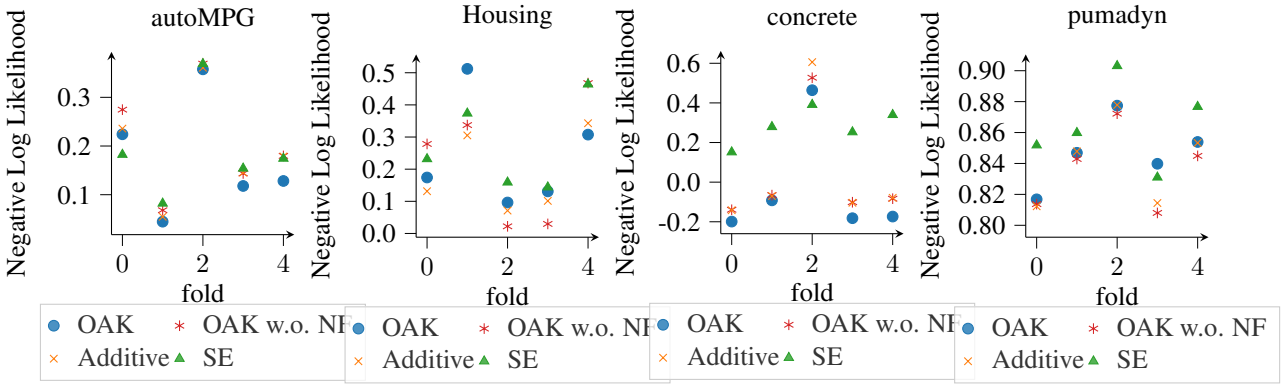


Figure 13. Negative log likelihood on regression datasets over 5 folds, lower is better. OAK w.o. NF stands for the OAK model without normalizing flow.

J.6. Comparison with Kernel in Duvenaud et al. (2011)

The kernel $\prod_d(1+\tilde{k}_d)$ restricts the lengthscales and variances of the kernels to be the same for lower and higher order terms, e.g., if two features are important in their main effect, the interaction between them will also be important, which may result in a less parsimonious model as higher order terms cannot be downweighted during inference. We have conducted experiments using this kernel for comparison, with results shown in Figure 18. We found the kernel is harder to optimize and numerically unstable, the model performance is similar but the resulting model is less parsimonious: e.g., Concrete dataset needs 3rd order terms (with normalized Sobol indices = 0.71, 0.16, 0.13 for 1st, 2nd and 3rd order respectively, as

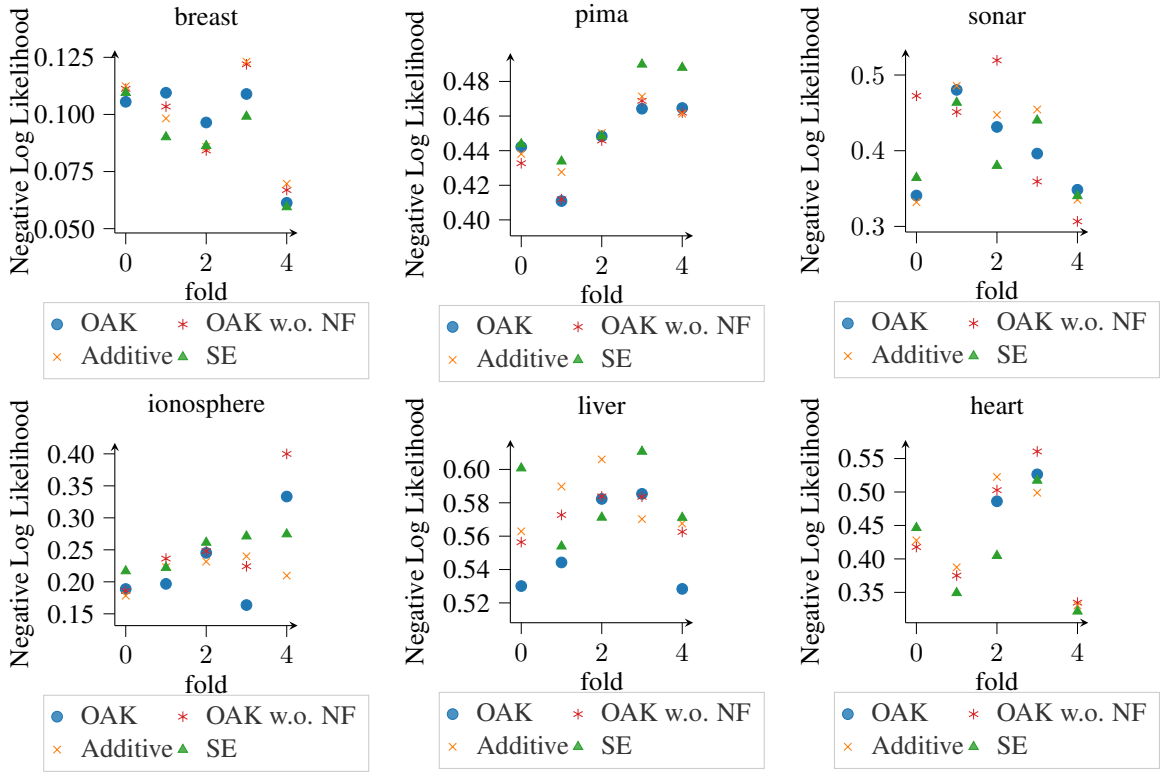


Figure 14. Negative log likelihood on classification datasets over 5 folds, lower is better. OAK w.o. NF stands for the OAK model without normalizing flow.

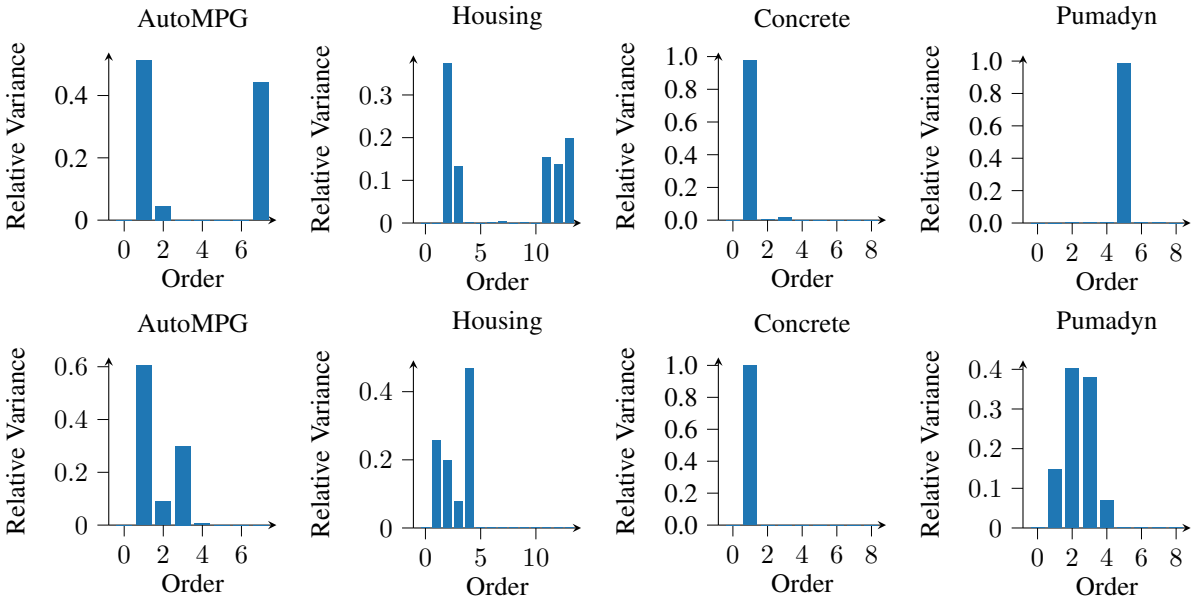


Figure 15. Normalized order variance hyperparameter on the UCI regression datasets. Top: kernel used in Duvenaud et al. (2011); bottom: OAK model. OAK requires lower dimensional orders of interactions with similar performance. Results are averaged over 5 folds.

opposed to 0.971, 0.026, 0.003 with OAK in Figure 4.

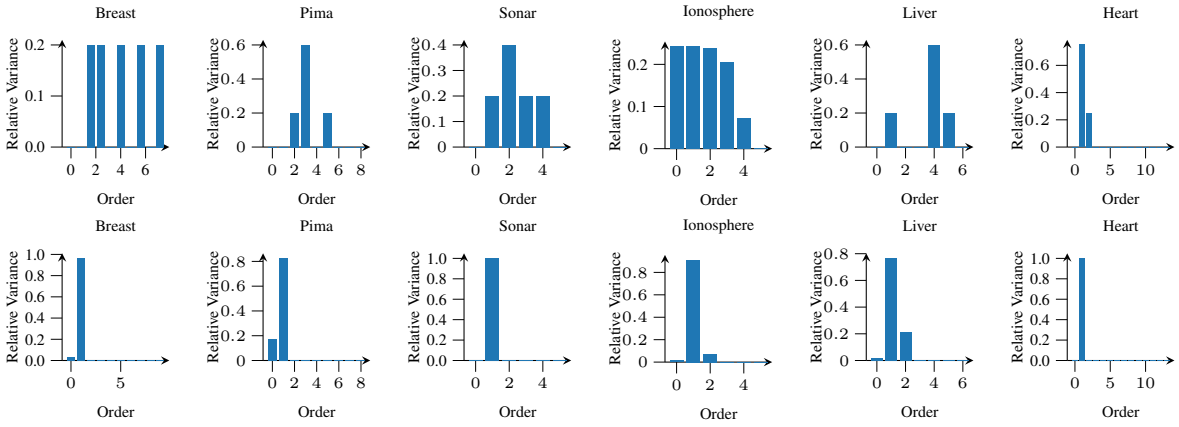


Figure 16. Normalized order variance hyperparameter on the UCI classification datasets. Top: kernel used in Duvenaud et al. (2011); bottom: OAK model. OAK requires lower dimensional orders of interactions with similar performance. We have truncated the maximum order of interaction to 4 for Sonar and Ionosphere datasets. Results are averaged over 5 folds.

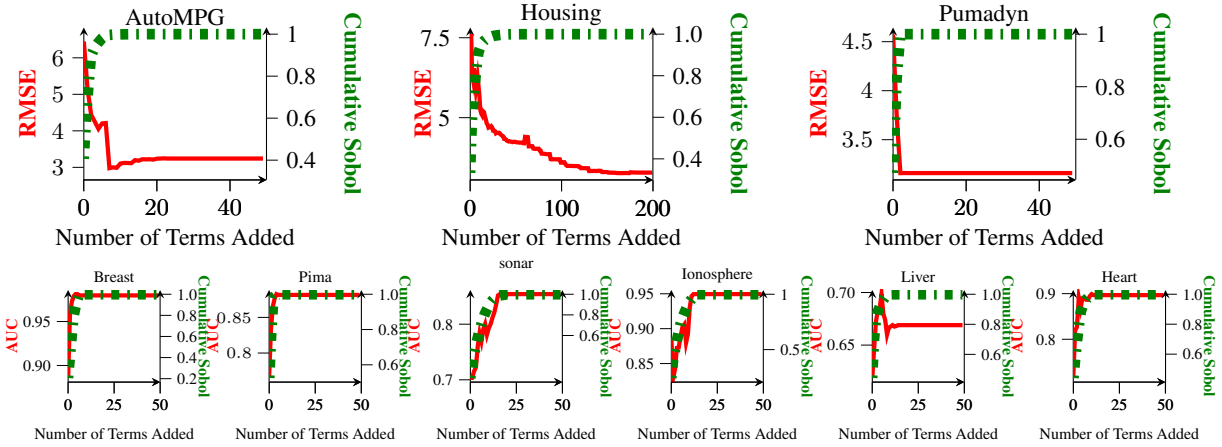


Figure 17. Model performance and cumulative Sobol index versus number of terms added ranked by the Sobol index, without normalizing flow. For regression problems (top), we use test RMSE as the evaluation metric. Note that we did not include result for the Concrete dataset because the NF was not sufficient to transform the data and we used the empirical measure for it in Figure 5: in this case the predictive performance was not affected, but the parsimony of the result (i.e. the number of terms needed to reach the same performance) was. For classification problems (bottom), we use test area-under-the-curve (AuC) metric. Red solid lines represent test RMSE (top) and test AuC (bottom), green dashed lines represent cumulative (normalized) Sobol index.

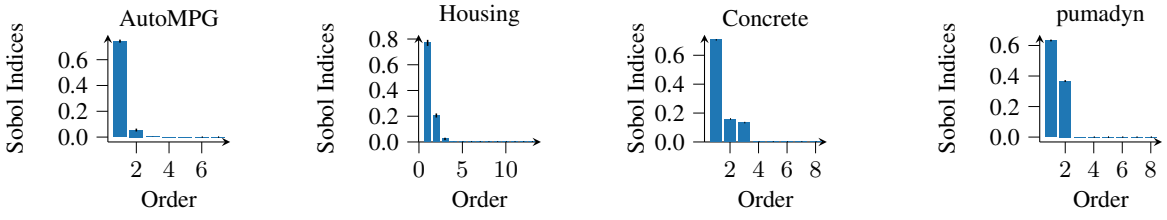


Figure 18. Cumulative Sobol Indices with kernel of the form $\prod_d(1+\tilde{k}_d)$ in Duvenaud et al. (2011) using constrained kernel. The model performance is similar to OAK but the resulting model tends to be less parsimonious: e.g., Concrete dataset needs 3rd order terms with normalized Sobol indices = 0.71, 0.16, 0.13 for 1st, 2nd and 3rd order respectively as opposed to 0.971, 0.026, 0.003 with OAK in Figure 4.

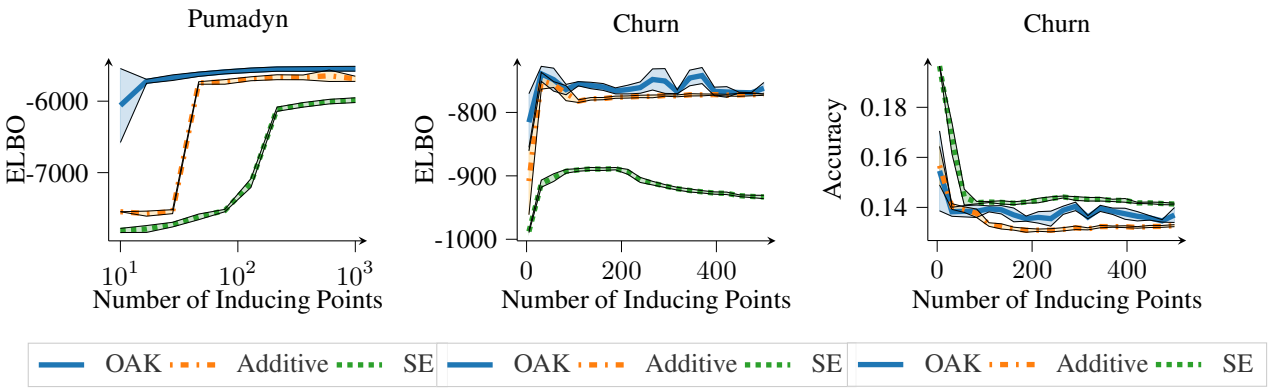


Figure 19. Model performance versus varying number of inducing points on the test set. Results are averaged over 5 repetitions for the Pumadyn dataset and 10 repetitions for the Churn dataset. Shaded area represents ± 1 standard deviation. Note that test ELBO is not always monotone on the Churn data, we attribute this to the difficulty of finding local optima.

J.7. Number of Inducing Points Needed

When kernels are combined through a product, the eigenspectrum is the outer-product of the spectra of the components (Corollary 3 in Burt et al. (2019)). This is what leads to the exponential scaling of the number of inducing points with the dimension of the problem, $M = \mathcal{O}(log^D N)$. When we add kernels together, the eigenspectrum is simply the concatenation of the spectrum of each component, so the resulting scaling is linear.

Additional experiments on number of inducing points needed for the pumadyn and Churn datasets can be found in Figure 19. The number of inducing points needed for OAK is smaller than that for the non-orthogonal model and the full GP model. Note that although the ELBO values are not directly comparable due to the normalizing flow used for some of the models, we can observe that the OAK model converges much faster than its counterparts.

K. Additional Benchmark Experiments

The evaluation results for the entire set of datasets[6] summarized in Table 1 can be found in Tables 5-8. Values outside $[-1000, 1000]$ are denoted as NaN. Results are averaged over 10 train-test splits, values in brackets represent one standard deviation.

[6]Data and code for other methods are taken from https://github.com/hughsalimbeni/bayesian_benchmarks.

dataset	N	D	OAK	Linear	SVGP	SVM	KNN	GBM	AdaBoost	MLP
boston	506	13	0.290(0.036)	0.444(0.044)	0.312(0.030)	0.267(0.037)	0.380(0.059)	0.282(0.020)	0.349(0.026)	0.299(0.030)
energy	768	8	0.036(0.010)	0.300(0.034)	0.048(0.005)	0.227(0.027)	0.218(0.029)	0.047(0.005)	0.191(0.006)	0.193(0.020)
naval	11934	14	0.164(0.313)	0.394(0.007)	0.004(0.001)	0.215(0.006)	0.104(0.006)	0.263(0.007)	0.885(0.017)	0.051(0.007)
power	9568	4	0.234(0.009)	0.267(0.008)	0.237(0.009)	0.234(0.009)	0.219(0.008)	0.226(0.008)	0.327(0.012)	0.236(0.009)
winered	1599	11	0.775(0.044)	0.808(0.046)	0.926(0.145)	0.768(0.055)	0.825(0.063)	0.762(0.046)	0.774(0.054)	0.773(0.055)
winewhite	4898	11	0.827(0.079)	0.847(0.033)	0.837(0.084)	0.768(0.021)	0.788(0.020)	0.768(0.020)	0.826(0.022)	0.763(0.023)
protein	45730	9	0.987(0.032)	0.850(0.004)	0.782(0.007)	0.764(0.008)	0.623(0.007)	0.768(0.007)	0.933(0.012)	0.707(0.022)
yacht	308	6	0.032(0.012)	0.608(0.048)	0.048(0.016)	0.419(0.092)	0.668(0.144)	0.044(0.014)	0.103(0.023)	0.244(0.051)
airfoil	1503	5	0.837(0.174)	0.721(0.047)	0.456(0.033)	0.486(0.038)	0.429(0.035)	0.387(0.043)	0.573(0.029)	0.412(0.041)
forest	517	12	1.030(0.100)	1.018(0.106)	0.995(0.025)	1.100(0.139)	1.117(0.142)	1.069(0.131)	1.092(0.093)	1.077(0.115)
parkinsons	195	23	0.373(0.140)	0.871(0.021)	0.635(0.021)	0.544(0.022)	0.384(0.024)	0.245(0.008)	0.587(0.020)	0.283(0.017)
stock	536	11	0.305(0.049)	0.286(0.025)	0.286(0.027)	0.465(0.136)	0.579(0.094)	0.348(0.058)	0.363(0.071)	0.308(0.025)
fertility	100	10	0.799(0.192)	0.900(0.229)	0.975(0.295)	0.975(0.250)	1.055(0.287)	1.032(0.225)	0.904(0.209)	1.020(0.233)
machine	209	7	0.281(0.044)	0.435(0.053)	0.398(0.048)	0.419(0.054)	0.417(0.076)	0.338(0.043)	0.368(0.037)	0.393(0.044)
pendulum	630	9	0.443(0.099)	0.862(0.164)	0.653(0.136)	0.654(0.188)	0.626(0.132)	0.772(0.110)	0.810(0.134)	0.659(0.140)
servo	167	4	0.312(0.069)	0.607(0.068)	0.299(0.074)	0.343(0.060)	0.454(0.070)	0.270(0.070)	0.383(0.062)	0.364(0.060)
wine	178	14	0.449(0.033)	0.564(0.029)	0.469(0.034)	0.440(0.041)	0.562(0.045)	0.461(0.031)	0.620(0.041)	0.436(0.038)
tamielectr	45781	3	1.001(0.005)	1.001(0.005)	1.001(0.005)	1.002(0.005)	1.099(0.007)	1.002(0.005)	1.002(0.005)	1.002(0.005)
kin40k	40000	8	0.581(0.019)	1.000(0.013)	0.682(0.016)	0.205(0.004)	0.392(0.005)	0.842(0.010)	0.939(0.013)	0.187(0.007)
gas	2565	128	0.254(0.078)	112.965(333.613)	0.182(0.041)	0.227(0.101)	0.119(0.037)	0.117(0.029)	0.313(0.025)	0.496(0.595)
keggdirect	48827	20	0.129(0.065)	nan	0.109(0.005)	0.102(0.002)	0.097(0.004)	0.094(0.003)	0.201(0.003)	0.199(0.318)
bike	17379	17	0.023(0.008)	0.517(0.008)	0.353(0.006)	0.262(0.008)	0.454(0.011)	0.020(0.001)	0.124(0.004)	0.065(0.008)
pol	15000	26	0.848(0.131)	0.736(0.011)	0.396(0.010)	0.335(0.006)	0.215(0.013)	0.256(0.008)	0.492(0.017)	0.151(0.007)
elevators	16599	18	0.379(0.007)	14.600(21.603)	0.394(0.007)	0.392(0.007)	0.602(0.016)	0.502(0.014)	0.776(0.014)	0.359(0.013)
avg			0.475	6.157	0.478	0.484	0.518	0.455	0.581	0.445
median			0.376	0.736	0.397	0.419	0.454	0.343	0.580	0.361
avg rank			3.583	6.625	4.083	4.208	4.958	3.208	5.750	3.583

Table 5. Test RMSE for regression tasks on additional benchmark datasets, lower is better.

dataset	N	D	OAK	Linear	SVGP	SVM	KNN	GBM	AdaBoost	MLP
boston	506	13	-0.122(0.157)	-0.644(0.066)	-0.281(0.058)	-0.157(0.083)	-0.467(0.134)	-0.637(0.250)	-0.388(0.095)	-0.248(0.140)
energy	768	8	1.923(0.308)	-0.220(0.114)	1.609(0.081)	0.038(0.159)	-0.021(0.254)	1.603(0.154)	0.235(0.035)	0.194(0.117)
naval	11934	14	1.932(1.525)	-0.489(0.017)	3.957(0.133)	0.120(0.028)	0.740(0.109)	-0.088(0.030)	-1.297(0.019)	1.561(0.144)
power	9568	4	0.030(0.037)	-0.098(0.031)	0.018(0.036)	0.034(0.038)	0.046(0.056)	0.066(0.042)	-0.304(0.037)	0.025(0.037)
winered	1599	11	-1.166(0.059)	-1.208(0.060)	-1.507(0.517)	-1.174(0.095)	-1.280(0.121)	-1.206(0.099)	-1.170(0.081)	-1.204(0.109)
winewhite	4898	11	-1.224(0.091)	-1.254(0.039)	-1.236(0.095)	-1.161(0.031)	-1.230(0.040)	-1.161(0.031)	-1.229(0.028)	-1.160(0.038)
protein	45730	9	-1.407(0.030)	-1.257(0.005)	-1.172(0.008)	-1.150(0.011)	-1.013(0.018)	-1.156(0.009)	-1.350(0.013)	-1.073(0.031)
yacht	308	6	1.320(1.503)	-0.929(0.083)	1.715(0.237)	-0.614(0.287)	-1.152(0.329)	-0.597(2.242)	0.799(0.351)	-0.090(0.318)
airfoil	1503	5	-1.395(0.600)	-1.096(0.070)	-0.650(0.072)	-0.711(0.093)	-0.693(0.149)	-0.496(0.139)	-0.865(0.054)	-0.548(0.119)
forest	517	12	-1.473(0.119)	-1.447(0.121)	-1.893(0.503)	-1.582(0.206)	-1.600(0.204)	-1.753(0.321)	-1.557(0.129)	-1.594(0.196)
parkinsons	195	23	-0.415(0.419)	-1.282(0.025)	-0.976(0.026)	-0.813(0.045)	-0.555(0.111)	-0.012(0.035)	-0.886(0.034)	-0.243(0.107)
stock	536	11	-0.199(0.111)	-0.175(0.079)	-0.173(0.078)	-1.090(0.975)	-0.975(0.287)	-1.100(0.687)	-0.486(0.331)	-0.344(0.164)
fertility	100	10	-1.244(0.239)	-1.376(0.362)	-1.461(0.425)	-1.676(0.735)	-1.631(0.538)	-3.890(1.808)	-1.608(0.608)	-2.800(1.437)
machine	209	7	-0.162(0.153)	-0.598(0.134)	-0.519(0.132)	-0.603(0.190)	-0.629(0.279)	-1.566(0.731)	-0.507(0.174)	-0.510(0.145)
pendulum	630	9	0.309(0.978)	-1.299(0.209)	-0.912(0.184)	-1.129(0.462)	-1.020(0.308)	-2.375(0.766)	-1.329(0.317)	-1.354(0.585)
servo	167	4	-0.402(0.522)	-0.929(0.098)	-0.265(0.211)	-0.400(0.245)	-0.783(0.304)	-0.418(0.660)	-0.513(0.225)	-0.432(0.207)
wine	178	14	-0.613(0.068)	-0.849(0.054)	-0.660(0.070)	-0.624(0.127)	-0.900(0.129)	-0.706(0.109)	-0.947(0.075)	-0.651(0.143)
tamielectr	45781	3	-1.461(0.117)	-1.420(0.005)	-1.420(0.005)	-1.421(0.005)	-1.561(0.010)	-1.422(0.005)	-1.420(0.005)	-1.421(0.005)
kin40k	40000	8	-0.874(0.030)	-1.419(0.013)	-1.034(0.022)	0.164(0.022)	-0.529(0.019)	-1.247(0.012)	-1.357(0.015)	0.253(0.037)
gas	2565	128	-0.166(0.260)	nan	0.292(0.103)	-0.051(0.379)	0.646(0.357)	-0.101(0.894)	-0.275(0.098)	-3.604(9.953)
keggdirect	48827	20	0.591(0.526)	nan	0.853(0.027)	0.853(0.030)	0.897(0.060)	0.945(0.033)	0.184(0.015)	-7.418(25.044)
bike	17379	17	2.416(0.383)	-0.759(0.015)	-0.379(0.017)	-0.085(0.033)	-0.688(0.040)	2.468(0.050)	0.666(0.037)	1.315(0.129)
pol	15000	26	-1.246(0.158)	-1.112(0.015)	-0.506(0.024)	-0.327(0.018)	0.052(0.095)	-0.058(0.033)	-0.710(0.033)	0.348(0.073)
elevators	16599	18	-0.450(0.023)	nan	-0.488(0.015)	-0.489(0.022)	-0.975(0.044)	-0.733(0.030)	-1.196(0.021)	-0.397(0.039)
avg			-0.229	-0.946	-0.295	-0.585	-0.638	-0.652	-0.730	-0.891
median			-0.409	-1.096	-0.512	-0.609	-0.738	-0.671	-0.875	-0.471
avg rank			5.583	3.625	5.042	4.833	3.917	4.292	3.583	5.125

Table 6. Test log likelihood for regression tasks on additional benchmark datasets, higher is better.

dataset	N	D	OAK	Linear	SVGP	SVM	KNN	GBM	AdaBoost	MLP
acute-infl	120	7	1.000(0.000)	1.000(0.000)	1.000(0.000)	1.000(0.000)	1.000(0.000)	1.000(0.000)	0.958(0.072)	1.000(0.000)
acute-neph	120	7	1.000(0.000)	1.000(0.000)	1.000(0.000)	1.000(0.000)	1.000(0.000)	1.000(0.000)	0.992(0.025)	1.000(0.000)
bank	4521	17	0.898(0.016)	0.891(0.018)	0.890(0.017)	0.891(0.013)	0.890(0.014)	0.900(0.011)	0.892(0.015)	0.893(0.012)
blood	748	5	0.740(0.007)	0.780(0.055)	0.787(0.051)	0.781(0.050)	0.767(0.041)	0.768(0.041)	0.776(0.046)	0.793(0.043)
chess-krvk	3196	37	0.960(0.021)	0.980(0.005)	0.980(0.006)	0.993(0.004)	0.959(0.007)	0.999(0.001)	0.999(0.001)	0.997(0.003)
congressio	435	17	0.616(0.050)	0.616(0.042)	0.605(0.050)	0.630(0.061)	0.568(0.067)	0.584(0.070)	0.582(0.056)	0.566(0.059)
conn-bench	208	61	0.990(0.019)	0.986(0.030)	0.976(0.038)	0.971(0.032)	0.900(0.054)	1.000(0.000)	1.000(0.000)	0.929(0.053)
credit-app	690	16	0.888(0.067)	0.849(0.051)	0.851(0.045)	0.833(0.036)	0.830(0.044)	0.967(0.025)	0.971(0.016)	0.858(0.037)
cylinder-b	512	36	0.752(0.060)	0.727(0.042)	0.735(0.034)	0.779(0.031)	0.785(0.053)	0.810(0.045)	0.738(0.030)	0.767(0.029)
echocardio	131	11	0.879(0.091)	0.850(0.126)	0.864(0.117)	0.850(0.098)	0.814(0.136)	0.843(0.070)	0.843(0.083)	0.843(0.114)
fertility	100	10	0.900(0.050)	0.900(0.063)	0.920(0.060)	0.920(0.060)	0.910(0.054)	0.860(0.092)	0.870(0.078)	0.890(0.070)
haberman-s	306	4	0.758(0.089)	0.755(0.087)	0.765(0.089)	0.745(0.065)	0.694(0.070)	0.713(0.101)	0.745(0.087)	0.745(0.070)
heart-hung	294	13	1.000(0.000)	0.997(0.010)	0.997(0.010)	0.970(0.023)	0.863(0.055)	1.000(0.000)	1.000(0.000)	0.990(0.015)
hepatitis	155	20	0.819(0.071)	0.794(0.097)	0.856(0.056)	0.844(0.075)	0.819(0.076)	0.812(0.079)	0.787(0.098)	0.844(0.075)
hill-valle	1212	101	0.483(0.048)	0.556(0.043)	0.484(0.043)	0.493(0.040)	0.507(0.031)	0.520(0.036)	0.517(0.038)	0.526(0.061)
horse-coli	368	26	0.824(0.039)	0.832(0.055)	0.824(0.057)	0.830(0.053)	0.781(0.052)	0.830(0.051)	0.792(0.042)	0.814(0.057)
ilpd-india	583	10	0.697(0.045)	0.702(0.050)	0.685(0.056)	0.681(0.072)	0.666(0.050)	0.649(0.039)	0.669(0.045)	0.649(0.042)
mammograph	961	6	0.830(0.024)	0.831(0.022)	0.836(0.023)	0.833(0.031)	0.802(0.038)	0.837(0.028)	0.827(0.028)	0.823(0.035)
molec-biol	106	58	0.964(0.060)	0.900(0.086)	0.900(0.103)	0.918(0.086)	0.927(0.089)	1.000(0.000)	1.000(0.000)	0.873(0.109)
monks-1	556	7	0.988(0.016)	0.629(0.046)	0.825(0.051)	0.845(0.042)	0.893(0.040)	0.995(0.008)	0.986(0.013)	0.973(0.022)
monks-2	601	7	0.685(0.062)	0.646(0.066)	0.652(0.055)	0.662(0.082)	0.754(0.070)	0.611(0.074)	0.567(0.038)	0.733(0.060)
monks-3	554	7	0.977(0.014)	0.714(0.067)	0.963(0.028)	0.955(0.018)	0.889(0.047)	0.988(0.011)	0.954(0.033)	0.968(0.024)
mushroom	8124	22	0.998(0.003)	0.953(0.006)	1.000(0.000)	1.000(0.000)	1.000(0.000)	1.000(0.000)	1.000(0.000)	1.000(0.000)
musk-1	476	167	1.000(0.000)	0.992(0.014)	0.988(0.019)	0.973(0.013)	0.904(0.038)	0.996(0.012)	0.996(0.012)	0.990(0.014)
musk-2	6598	167	0.998(0.004)	1.000(0.000)	1.000(0.000)	0.998(0.002)	0.977(0.006)	1.000(0.000)	1.000(0.000)	1.000(0.000)
oocytes_me	1022	42	0.846(0.024)	0.784(0.022)	0.838(0.020)	0.779(0.022)	0.722(0.035)	0.780(0.025)	0.755(0.040)	0.841(0.023)
oocytes_tr	912	26	0.837(0.015)	0.774(0.035)	0.822(0.039)	0.823(0.034)	0.728(0.062)	0.817(0.035)	0.778(0.036)	0.830(0.027)
ozone	2536	73	0.973(0.011)	0.972(0.007)	0.972(0.009)	0.972(0.010)	0.971(0.011)	0.970(0.009)	0.973(0.008)	0.970(0.008)
parkinsons	195	23	0.985(0.023)	0.795(0.085)	0.895(0.099)	0.890(0.062)	0.935(0.045)	0.970(0.046)	0.930(0.046)	0.930(0.051)
avg			0.872	0.835	0.859	0.857	0.836	0.870	0.859	0.863
median			0.898	0.832	0.864	0.850	0.863	0.900	0.892	0.873
avg rank			5.569	4.224	4.741	4.500	2.983	5.224	4.207	4.552

Table 7. Test accuracy for classification tasks on additional benchmark datasets, higher is better.

dataset	N	D	OAK	Linear	SVGP	SVM	KNN	GBM	AdaBoost	MLP
acute-infl	120	7	-0.003(0.000)	-0.057(0.007)	-0.001(0.000)	-0.018(0.001)	-0.000(0.000)	-0.000(0.000)	-0.057(0.091)	-0.032(0.008)
acute-neph	120	7	-0.003(0.003)	-0.032(0.009)	-0.001(0.000)	-0.019(0.001)	-0.000(0.000)	-0.000(0.000)	-0.085(0.256)	-0.017(0.004)
bank	4521	17	-0.248(0.024)	-0.271(0.029)	-0.262(0.027)	-0.286(0.029)	-1.143(0.224)	-0.235(0.020)	-0.646(0.002)	-0.282(0.035)
blood	748	5	-0.491(0.023)	-0.469(0.071)	-0.469(0.070)	-0.505(0.070)	-1.861(0.769)	-0.524(0.090)	-0.677(0.005)	-0.473(0.073)
chess-krvk	3196	37	-0.078(0.041)	-0.056(0.009)	-0.051(0.011)	-0.020(0.008)	-0.232(0.088)	-0.010(0.015)	-0.010(0.007)	-0.020(0.018)
congressio	435	17	-0.655(0.040)	-0.697(0.100)	-0.650(0.041)	-0.666(0.026)	-2.172(1.056)	-0.699(0.064)	-0.687(0.003)	-0.803(0.175)
conn-bench	208	61	-0.040(0.026)	-0.090(0.070)	-0.084(0.092)	-0.077(0.040)	-0.208(0.073)	-0.000(0.000)	-0.000(0.000)	-0.189(0.117)
credit-app	690	16	-0.280(0.126)	-0.365(0.083)	-0.369(0.085)	-0.377(0.070)	-1.182(0.552)	-0.111(0.074)	-0.583(0.008)	-0.363(0.080)
cylinder-b	512	36	-0.475(0.055)	-0.533(0.059)	-0.532(0.033)	-0.463(0.045)	-0.888(0.375)	-0.392(0.049)	-0.654(0.007)	-0.530(0.149)
echocardio	131	11	-0.358(0.170)	-0.394(0.185)	-0.376(0.157)	-0.423(0.167)	-1.107(1.274)	-0.444(0.273)	-0.584(0.024)	-0.385(0.202)
fertility	100	10	-0.380(0.134)	-0.341(0.213)	-0.296(0.115)	-0.298(0.123)	-1.546(1.799)	-0.561(0.484)	-0.633(0.039)	-0.362(0.239)
haberman-s	306	4	-0.532(0.099)	-0.531(0.093)	-0.530(0.106)	-0.540(0.094)	-1.468(1.395)	-0.570(0.157)	-0.679(0.009)	-0.540(0.114)
heart-hung	294	13	-0.007(0.002)	-0.044(0.016)	-0.008(0.016)	-0.063(0.035)	-1.088(0.817)	-0.000(0.000)	-0.000(0.000)	-0.046(0.023)
hepatitis	155	20	-0.414(0.105)	-0.389(0.100)	-0.346(0.065)	-0.352(0.077)	-1.306(0.798)	-0.531(0.159)	-0.570(0.046)	-0.362(0.130)
hill-valle	1212	101	-0.694(0.001)	-0.650(0.013)	-0.693(0.000)	-0.694(0.001)	-1.498(0.294)	-0.708(0.027)	-0.693(0.004)	-0.675(0.013)
horse-coli	368	26	-0.406(0.077)	-0.455(0.101)	-0.433(0.085)	-0.422(0.074)	-1.609(0.776)	-0.388(0.104)	-0.666(0.007)	-0.517(0.154)
ilpd-india	583	10	-0.555(0.028)	-0.548(0.040)	-0.548(0.036)	-0.605(0.053)	-1.695(0.679)	-0.633(0.078)	-0.636(0.011)	-0.582(0.056)
mammograph	961	6	-0.386(0.048)	-0.419(0.041)	-0.406(0.043)	-0.403(0.046)	-1.278(0.543)	-0.386(0.056)	-0.665(0.005)	-0.409(0.063)
molec-biol	106	58	-0.149(0.172)	-0.211(0.138)	-0.203(0.136)	-0.196(0.171)	-0.283(0.100)	-0.000(0.000)	-0.000(0.000)	-0.363(0.186)
monks-1	556	7	-0.027(0.017)	-0.618(0.044)	-0.307(0.066)	-0.389(0.083)	-0.546(0.235)	-0.040(0.012)	-0.569(0.013)	-0.159(0.038)
monks-2	601	7	-0.549(0.063)	-0.648(0.044)	-0.638(0.048)	-0.589(0.056)	-0.676(0.398)	-0.653(0.085)	-0.679(0.007)	-0.505(0.056)
monks-3	554	7	-0.067(0.037)	-0.453(0.089)	-0.091(0.054)	-0.149(0.059)	-0.543(0.255)	-0.048(0.022)	-0.638(0.008)	-0.095(0.033)
mushroom	8124	22	-0.009(0.021)	-0.135(0.009)	-0.002(0.001)	-0.000(0.000)	-0.000(0.000)	-0.003(0.000)	-0.483(0.006)	-0.001(0.000)
musk-1	476	167	-0.015(0.011)	-0.058(0.039)	-0.056(0.054)	-0.079(0.035)	-0.340(0.206)	-0.045(0.136)	-0.115(0.345)	-0.079(0.073)
musk-2	6598	167	-0.008(0.015)	-0.004(0.001)	-0.001(0.000)	-0.006(0.006)	-0.127(0.059)	-0.002(0.005)	-0.004(0.013)	-0.002(0.000)
oocytes_me	1022	42	-0.368(0.078)	-0.454(0.024)	-0.391(0.049)	-0.467(0.032)	-1.541(0.433)	-0.459(0.028)	-0.675(0.007)	-0.397(0.066)
oocytes_tr	912	26	-0.382(0.034)	-0.488(0.052)	-0.416(0.054)	-0.408(0.051)	-1.224(0.552)	-0.417(0.052)	-0.674(0.003)	-0.370(0.044)
ozone	2536	73	-0.107(0.035)	-0.087(0.025)	-0.082(0.023)	-0.098(0.028)	-0.372(0.150)	-0.093(0.025)	-0.523(0.043)	-0.128(0.056)
parkinsons	195	23	-0.065(0.050)	-0.320(0.094)	-0.207(0.102)	-0.253(0.090)	-0.150(0.043)	-0.256(0.399)	-0.439(0.047)	-0.196(0.041)
avg			-0.267	-0.338	-0.291	-0.306	-0.899	-0.283	-0.459	-0.306
median			-0.280	-0.389	-0.307	-0.352	-1.088	-0.256	-0.584	-0.362
avg rank			5.862	4.276	5.931	4.690	2.138	5.379	2.897	4.828

Table 8. Test log likelihood for classification tasks on additional benchmark datasets, higher is better.