Combining additivity and active subspaces for high-dimensional Gaussian process modeling

Mickael Binois, Victor Picheny

Combining additivity and active subspaces for high-dimensional Gaussian process modeling

Mickaël Binois* Victor Picheny†

February 5, 2024

Abstract

Gaussian processes are a widely embraced technique for regression and classification due to their good prediction accuracy, analytical tractability and built-in capabilities for uncertainty quantification. However, they suffer from the curse of dimensionality whenever the number of variables increases. This challenge is generally addressed by assuming additional structure in the problem, the preferred options being either additivity or low intrinsic dimensionality. Our contribution for high-dimensional Gaussian process modeling is to combine them with a multi-fidelity strategy, showcasing the advantages through experiments on synthetic functions and datasets.

1 Introduction

As a surrogate modeling option, Gaussian processes (GPs), also known as kriging, enjoy widespread use across applied scientific domains, including engineering, machine learning and physics (see e.g., Williams & Rasmussen, 2006; Gramacy, 2020). Appreciated for their efficiency on small datasets, GPs offer a full predictive distribution in closed form and are readily accessible from numerous software packages. Nonetheless, the case of many input variables remains one of the most challenging topic in GP modeling, and particularly in its use within Bayesian optimization (BO) (see e.g., Garnett, 2023). The root of the issue is that the typical covariance kernels employed in applications, e.g., the squared exponential or Matérn ones, hinge on Euclidean and absolute distances between data points, as discussed, e.g., by Wilson et al. (2016). As the dimension increases, so do the distances between designs. Hence, in high-dimensional spaces, GP predictions predominantly operate in the

*Corresponding author: Inria, Université Côte d'Azur, CNRS, LJAD, Sophia Antipolis, France mickael.binois@inria.fr

†Secondmind, Cambridge, UK

extrapolation regime, where the selection of a trend (or mean function) is critical, see e.g., Journel & Rossi (1989).

Among possible structural assumptions to scale with respect to the number of variables, three main categories have emerged. The first one is to identify the most important variables and then reduce the problem dimensionality. This adaptive variable selection is applied for instance by Cao et al. (2022), while remaining efficient in both the number of variables and dimensions through the use of the Vecchia approximation, which restricts the conditioning set of each observations to only a few designs. A second, more general approach, entails assuming that the problem has a low intrinsic dimensionality, meaning that the variation of the function is concentrated on a few directions only. This is also referred to as using linear embeddings Wang et al. (2013); Garnett et al. (2014) or active subspaces (AS, Constantine et al., 2014; Eriksson et al., 2019). Hence, both approach avoid dealing directly with high-dimensional distances. A third perspective is to consider additive decompositions of the function, where components involve only a few variables, thereby limiting the degree of interaction between variables, see e.g., Duvenaud et al. (2011); Durrande et al. (2012); Rolland et al. (2018); Lu et al. (2022). Ginsbourger et al. (2016), decomposes a regular product Gaussian kernel into ANOVA terms allowing the resulting GP to undergo a similar decomposition. The originality is to separate elements into additive and ortho-additive components (i.e., that capture all the non-additive parts). For a more comprehensive review of high-dimensional GPs and BO, we refer to Malu et al. (2021); Binois & Wycoff (2022).

Here we focus on these two promising avenues that have been explored separately: the use of linear embeddings and of additive models. Each approach comes with its own set of strengths and limitations. Linear embeddings offer great scalability and allow capturing complex interactions, but only as long as the intrinsic dimension is low. Besides, the intrinsic dimension is generally not know before hand and the method is very sensitive to its estimation, as it is to the choice of the embedding. Additive models typically capture very well the main trends from high-dimensional data. However, as the number of possible interactions terms makes the full inference combinatorially intractable, with too many hyperparameters, it is common practice to limit the number of interactions to a manageable few, which severely limits the expressivity of those models. Additional concerns may arise in BO due to the vanishing variance at unexplored locations Durrande et al. (2012).

Given the distinct relative advantages of each approach – namely the scalability and interpretability offered by additive GPs and the ability to capture high-order interactions through AS-based GPs – we aim to propose a hybrid model that incorporates the strengths

of both. As each can capture the features of the other, with high-order interactions or large intrinsic dimensionality, combining them is not trivial. To address identifiability issues, we introduce orthogonality by adopting a multi-fidelity approach, typically used when inexpensive yet coarse approximations of the target function are available, see e.g., Kennedy & O'Hagan (2000); Brevault et al. (2020). In a nutshell, our model will consider two fidelity levels: a coarse level corresponding to a first order additive model and a high-fidelity level by a GP on an active subspace whose dimension is learned.

Our contributions are the following:

- We concentrate on two efficient structural assumptions for high-dimensional GP modeling, namely additivity and linear embeddings. We further detail their respective strengths and weaknesses, plus the complexity of a naive combination.
- We develop a multi-fidelity approach designed to efficiently perform this combination, capturing additive and linear embedding contributions.
- For AS-based GPs, we discuss inference of the intrinsic dimension within one- or two-stage methods.
- We conduct a thorough comparison of our multi-fidelity method against baselines within an extensive benchmark comprising test functions and datasets. Our findings confirm that the multi-fidelity approach improves over a standard GP when either additivity or active subspaces are present. Importantly, the performance does not degrade when such structures are absent.

2 Background

We want to fit a Gaussian process model of $f : \mathbf{X} \subset \mathbb{R}^d \to \mathbb{R}$ when d is relatively large. What *large* means depends on the dataset size and the complexity of the problem at hand. In the derivative free black-box context, it is generally considered that ten variables problems already fall within the realm of high dimensionality. We briefly outline Gaussian process regression, the additive and linear embedding versions, before introducing the multi-fidelity model.

2.1 Gaussian Processes Regression

Given $n \in \mathbb{N}$ input designs $\mathbf{x}^{(i)} \in \mathbf{X}$ with corresponding observations $f(\mathbf{x}^{(i)}) = y_i$ (possibly noisy), GPs are a form of spatial modeling that only depends on a mean and a covariance function. Typically, the mean function is taken to be zero, and all the modeling effort is placed on the covariance function k. From this GP prior, the posterior distribution is another GP and the prediction at any $\mathbf{x}$ follows: $Y(\mathbf{x})|(\mathbf{x}, y_i)_{1\leq i\leq n} \sim \mathcal{N}\left(m_n(\mathbf{x}), s_n^2(\mathbf{x})\right)$ where, see e.g., Williams & Rasmussen (2006); Gramacy (2020):

$$
\begin{aligned}
m_n(\mathbf{x}) &= \mathbf{k}(\mathbf{x})^\top \mathbf{K}^{-1}\mathbf{y}, \\
s_n^2(\mathbf{x}) &= k(\mathbf{x},\mathbf{x}) - \mathbf{k}(\mathbf{x})^\top \mathbf{K}^{-1}\mathbf{k}(\mathbf{x})
\end{aligned}
$$

with $\mathbf{y} := (y_1, \ldots, y_n)$, $\mathbf{k}(\mathbf{x}) := (k(\mathbf{x}, \mathbf{x}^{(i)}))_{1\leq i\leq n}$, $\mathbf{K} := (k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) + \tau^2 \mathbf{1}_{i=j})_{1\leq i,j\leq n}$. τ^2 is the noise hyperparameter, when assuming $y_i = f(\mathbf{x}^{(i)}) + \varepsilon_i$, with $\varepsilon_i \sim \mathcal{N}(0, \tau^2)$.

The covariance kernel function must be a positive definite function. In practice, parameterized families such as the Gaussian and Matérn covariances are employed, see e.g., Williams & Rasmussen (2006). As an example, the Matérn 5/2 kernel in product form writes: $k(\mathbf{x}, \mathbf{x}') = \sigma^2 \prod_{i=1}^{d} k_i(x_i, x_i')$ with $k_i(x_i, x_i') = \left(1 + \sqrt{5}h_i/\theta_i + 5h_i^2/(3\theta^2)\right) \exp\left(-\sqrt{5}h_i/\theta\right)$. For inferring the hyperparameters, we rely on the log-likelihood: $\log L := -n/2 \log(2\pi) - 1/2 \log|\mathbf{K}| - 1/2 \mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}$. When the variance parameter σ^2 can be factorized, i.e. $\mathbf{K} = \sigma^2 \mathbf{R}$, with $\mathbf{R}$ the correlation matrix, its estimator is available in closed-form: $\hat{\sigma}_n^2 := n^{-1}\mathbf{y}^\top \mathbf{R}^{-1}\mathbf{y}$, while the other hyperparameters are obtained by maximizing the concentrated log-likelihood: $\log \tilde{L} := -n/2 \log(2\pi) - n/2 \log(\hat{\sigma}_n^2) - n/2 \log|\mathbf{R}| - n/2$.

A typical extension of GPs to offer much better scalability with data size is to follow the Sparse Variational GP (SVGP) framework (Titsias, 2009; Hensman et al., 2013). While not considered in this paper, our model would naturally apply to this framework.

2.2 Additive Model

Unlike tensor product covariance kernels whose values quickly decrease to zero, impacting the covariance values hence the modeling ability, the tensor sum counterparts do not suffer from this problem. This latter form of covariance amounts to considering additive models, that is, decompositions of the original function into several components. The general model writes $f(\mathbf{x}) \approx \mu + \sum_{i=1}^{M} g_i(\mathbf{x}_{A_i})$ with component functions g_i acting on subsets of variables A_i, plus a constant term μ. These subsets can simply be the original variables Neal (1997); Plate

(1999); Duvenaud et al. (2011); Durrande et al. (2012), disjoint groups Kandasamy et al. (2015); Gardner et al. (2017) or more general subsets of variables Rolland et al. (2018).

The sum form of the covariance, i.e., $k_A(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{d} k_i(x_i, x_i')$, translates in the model, where the mean becomes the sum of component-wise means $m_{n,A}(\mathbf{x}) = \mathbf{k}_A(\mathbf{x})^\top \mathbf{K}_A^{-1}\mathbf{y} = \sum_{i=1}^{d} \mathbf{k}_i(x_i)\mathbf{K}_A^{-1}\mathbf{y} = \sum_{i=1}^{d} m_{n,i}(x_i)$ with $\mathbf{k}_i(x_i) := (k_i(x_i, x_i^{(j)}))_{1\leq j\leq n}$. It becomes useful for visualization and interpretation, e.g., with main effect plots. As for the predictive variance, it does not have a similar decomposition but it can can be zero at unobserved locations, unless noise is present, see, e.g., Durrande et al. (2012).

Inference can involve learning variance and scale parameters for every component kernel, plus eventually selecting interaction order, with up to 2^d components. To help inference, centering the various terms is usually preferred to avoid non-identifiability Durrande et al. (2012); Lu et al. (2022). Orthogonality constraints can be further added between the terms, leading to functional ANOVA decomposition of the original function, see e.g., Muehlenstaedt et al. (2012); Durrande et al. (2013); Ginsbourger et al. (2016).

2.3 Active Subspace Methods

By not imposing the variables to match the original variables in dimension reduction, one can rather attempt to learn the most important directions of variation of f: $f(\mathbf{x}) \approx g(\mathbf{A}^\top \mathbf{x})$ with $\mathbf{A}$ a $d \times r$ matrix, $1 \leq r \leq d$ and preferably $r \ll d$. Learning this linear embedding encoded in $\mathbf{A}$ is possible with different strategies. Elements of $\mathbf{A}$ can be treated as regular hyperparameters Garnett et al. (2014); Tripathy et al. (2016); Letham et al. (2020), or they can be random Wang et al. (2013); Nayebi et al. (2019), relying on the stability of the random projection for the L_2 norm.

When looking at directions where f varies the most, the so-called active subspace Constantine (2015), $\mathbf{A}$ is defined (up to a rotation) as the largest r eigen vectors of the matrix $\mathbf{C} := \int_{\mathbf{X}} \nabla(f(\mathbf{x}))^\top \nabla(f(\mathbf{x}))\lambda(d\mathbf{x})$ where λ is usually the Lebesgue measure on hypercubic domains. Without the gradient of f, $\mathbf{A}$ may be estimated via compressed sensing, partial least squares, principal component analysis, see e.g., Carpentier & Munos (2012); Djolonga et al. (2013); Bouhlel et al. (2016); Raponi et al. (2020). For a GP, its AS matrix $\mathbf{C}^{(n)}$ can be directly computed, as shown by Wycoff et al. (2021) and detailed in Appendix A. These AS approaches usually involve first learning a high-dimensional GP to estimate $\mathbf{A}$, before fitting a low dimensional GP in the reduced space, see e.g., Tripathy et al. (2016). In Appendix A, we also show how to learn directly the low dimensional GP.

2.4 Multi-fidelity

Be it a number of Monte Carlo iterations, a mesh or a training set size, the accuracy of a simulator experiment is often tunable. Accordingly, GP models have been adapted to take into account these various levels of fidelity, see e.g., Kennedy & O'Hagan (2000); Forrester et al. (2008); Le Gratiet & Garnier (2014); Tighineanu et al. (2022). We only review the two levels case here, coarse (resp. fine) level denoted by C (resp. E), with the auto-regressive (AR) model: $f_E(\mathbf{x}) = \rho f_C(\mathbf{x}) + \delta(\mathbf{x})$, $\delta(\cdot) \perp f_C(\cdot)$. This model, proposed by Kennedy & O'Hagan (2000), assumes that $\forall \mathbf{x} \neq \mathbf{x}'$, $\mathrm{Cov}\left[Y_E(\mathbf{x}), Y_C(\mathbf{x}') | Y_C(\mathbf{x})\right] = 0$, i.e., that nothing more can be learned for $Y_E(\mathbf{x})$ from the coarse model if $Y_C(\mathbf{x})$ is known.

Denote the n_C observations $\mathbf{y}_C$ (resp. $\mathbf{y}_E$) at $\mathbf{X}_C := \left(\mathbf{x}_C^{(1)}, \ldots, \mathbf{x}_C^{(n_C)}\right)$ (resp. $\mathbf{X}_E$). Given the following covariances:

$$
\begin{aligned}
\mathrm{Cov}\left[Y_C(\mathbf{x}), Y_C(\mathbf{x}')\right] &= k_C(\mathbf{x}, \mathbf{x}'), \\
\mathrm{Cov}\left[Y_E(\mathbf{x}), Y_C(\mathbf{x}')\right] &= \rho k_C(\mathbf{x}, \mathbf{x}'), \\
\mathrm{Cov}\left[Y_E(\mathbf{x}), Y_E(\mathbf{x}')\right] &= \rho^2 k_C(\mathbf{x}, \mathbf{x}') + k_E(\mathbf{x}, \mathbf{x}'),
\end{aligned}
$$

the corresponding predictive equations for the zero mean version are given by (see, e.g., Kennedy & O'Hagan (2000); Forrester et al. (2008) for the derivation):

$$
\begin{aligned}
m_{n,E}(\mathbf{x}) &= \tilde{\mathbf{k}}(\mathbf{x})^\top \tilde{\mathbf{K}}^{-1} \tilde{\mathbf{y}}, \\
s^2_{n,E}(\mathbf{x}) &= \rho^2 k_C(\mathbf{x}, \mathbf{x}) + k_E(\mathbf{x}, \mathbf{x}) - \tilde{\mathbf{k}}(\mathbf{x})^\top \tilde{\mathbf{K}}^{-1} \tilde{\mathbf{k}}(\mathbf{x})
\end{aligned}
\tag{1}
$$

with $\tilde{\mathbf{k}}(\mathbf{x})^\top = [\rho k_C(\mathbf{X}_C, \mathbf{x}), \rho^2 k_C(\mathbf{X}_E, \mathbf{x}) + k_E(\mathbf{X}_E, \mathbf{x})]$,
$\tilde{\mathbf{K}} = \begin{bmatrix} k_C(\mathbf{X}_C, \mathbf{X}_C) & \rho k_C(\mathbf{X}_C, \mathbf{X}_E) \\ \rho k_C(\mathbf{X}_E, \mathbf{X}_C) & \rho^2 k_C(\mathbf{X}_E, \mathbf{X}_E) + k_E(\mathbf{X}_E, \mathbf{X}_E) \end{bmatrix}$ and $\tilde{\mathbf{y}}^\top = [\mathbf{y}_C, \mathbf{y}_E]$.

For inference, the low fidelity model is independently trained first, then the fine level hyperparameters (including ρ) are obtained based on $\mathbf{d} := \mathbf{y}_E - \rho \mathbf{y}_C(\mathbf{X}_E)$. If the designs of experiments between fidelity levels are nested, i.e., $\mathbf{X}_E \subseteq \mathbf{X}_C$, the difference between levels can be directly evaluated. Otherwise, the difference can be computed based on the predictive mean Forrester et al. (2008); Sacher et al. (2021), replacing $\mathbf{y}_C(\mathbf{X}_E)$ by its prediction. Further details are given in Appendix B.

A recursive formulation of Eq. (1) is available to reduce the computational effort, see e.g., Le Gratiet & Garnier (2014), but equivalence holds only in the noiseless setting, see Appendix B. Subsequently we introduce our proposed multi-fidelity combination of additive

and linear embedding models, tailored to tackle high-dimensional problems.

3 Multi-fidelity for High-dimensional Modeling

Our goal is to combine the advantages of both additive and linear embedding models, without further complexifying inference. Ideally, each component would capture distinct features of the high-dimensional black-box, thereby enhancing the overall model performance.

3.1 Combination Options

There are presumably many options to combine models. A straightforward idea would be to simply sum the two types of models, but this raises identifiability issues. Without the AS assumption, this is the model proposed by Plate (1999) for visualization, by gradually modifying the degree of additivity. Another such idea is first to apply a rotation with AS as a preprocessing step, e.g., as in Wycoff et al. (2022), before applying an additive model on the inactive directions. The drawbacks here are the loss of interpretability of the additive model in the original variables and a potential lack of interpolation if the additivity assumption does not hold. The converse is to learn a linear embedding directly on the residuals of an additive model, which result in a challenging inference problem if done in one step. One workaround to include orthogonality conditions would be to follow Lenz (2013); Ginsbourger et al. (2016) with an orthogonal decomposition between additive and ortho-additive components, before applying AS on the latter. While appealing, this decomposition remains based on the projection of a single high-dimensional tensor product kernel. Furthermore, the independent integral derivations of ortho-additive and AS components do not seamlessly carry over when combined.

3.2 A Multi-fidelity Approach

To maintain an orthogonality condition for identifiability, we propose to rely on the one enjoyed by the multi-fidelity model. Indeed, this model regresses the coarse model when no data is available, in order to improve extrapolation–the predominant prediction regime characteristic of high-dimensional problems. We opt for a first order additive model as the coarse level and a linear embedding model as the finer one. This choice is the most natural since (first order) additivity is a restrictive yet data-efficient assumption. The linear embedding then is able to learn the remaining high orders of interaction, allowing flexibility

in the choice of the embedding dimension r. In the case $r = d$, a regular GP is fit on the residuals between additive model prediction and data, still on the rotated initial input space, thus maintaining interpolation in the noiseless case. This rotation is shown to be helpful as a pre-processing step Wycoff et al. (2022). If the additive model is a good approximation, the remaining variance of the GP on the residuals will be smaller. This thus alleviates the inflation of the variance in high dimension stemming from the behavior of distances, causing over-exploration as often observed in high-dimensional BO, see e.g., Eriksson et al. (2019). The converse, taking an AS model as coarse model is less relevant as it will also capture additive components while a finer additive model may not be able to interpolate deterministic data. Plus, again, the additive model would be on a rotated input space, losing interpretability and convenience. Our proposed model thus writes:

$$\begin{cases} Y_E(\mathbf{x}) = \rho Y_C(\mathbf{x}) + \delta(\mathbf{A}\mathbf{x}) \\ Y_C(\mathbf{x}) \perp \delta(\mathbf{A}\mathbf{x}) \end{cases} \tag{2}$$

In practice there is solely one set of (fine level) evaluations $\mathbf{y}$. To obtain coarse level values and $\mathbf{d}$, we simply take the predictions provided by the additive model: $\mathbf{y}^{(C)} = m_n^{(C)}(\mathbf{X}_C)$. An issue that may arise with this scheme is when an additive model fits the data perfectly. This scenario is more likely to occur when the training dataset is small, or the dimension large. Figure 1 illustrates an example using the non-additive Branin function. There are several options to cope with this issue, which can be detected by comparing the noise variance to the process variance. One is to restrict the range of the lengthscale (e.g., using prior knowledge). Another one is to withhold some designs from the additive model. That way, if the prediction by the additive model is not accurate, then it will be corrected at the fine level. Lastly, it remains to build the finer model on the residuals of the prediction by the coarse level. When fitting a linear embedding model, one key question is the selection of the dimension. We choose to rely on the likelihood to do so.

3.3 Proposed Instantiation

We detail the construction of model (2) in Algorithm 1. In Step 2, we restrict ourselves to using a first order additive model, avoiding higher order terms selection. From Steps 3 to 6, in case the noise variance of the additive model τ_C^2 is less than one percent of the additive process variance, $\sigma_{n,C}^2 = \sum_{i=1}^{d} \alpha_i$, then the additive model is replaced by one trained on a fraction p of the data. In Step 7, the low fidelity data is obtained by predicting with the

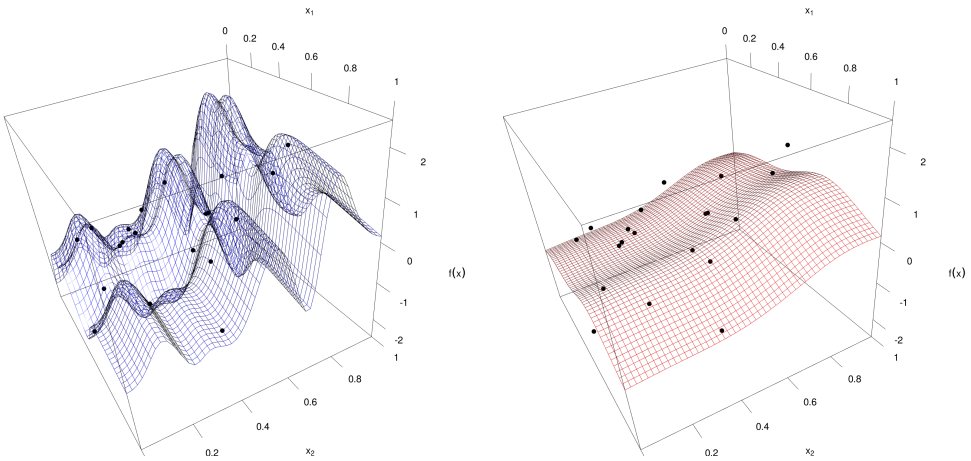

Figure 1: Additive model prediction surfaces on 20 points from the Branin function, interpolating (left) or approximating (right).

Algorithm 1 Pseudo-code for multi-fidelity high dim GP

1: **Input:** $\mathbf{X}_E = \mathbf{X}_C$, $\mathbf{y}$, p (e.g., 0.8)
2: Train an additive model Y_C on $(\mathbf{X}_C, \mathbf{y})$
3: **if** $\tau_C^2 \leq 0.01 \times \sum_1^d \alpha_i$ **then**
4: Sample $n_0 = p \times n$ data points from $\mathbf{x}_{1:n}$, $\mathbf{y}$ and remove the rest from $\mathbf{X}_C$ and $\mathbf{y}^{(C)}$.
5: Train an additive model Y_C on $(\mathbf{X}_C, \mathbf{y}^{(C)})$.
6: **end if**
7: Predict the response of Y_C at $\mathbf{X}_E$: $m_n^{(C)}(\mathbf{X}_E)$.
8: Train a multi-fidelity GP from the residual data: $\mathbf{d} = \mathbf{y} - \rho m_n^{(C)}(\mathbf{X}_E)$.
9: Estimate the corresponding AS matrix $\mathbf{C}^{(n)}$.
10: Train an AS multi-fidelity GP, varying the number of dimensions kept r.
11: **Output:** Trained multi-fidelity model.

additive model. The remaining steps are dedicated to learning the linear embedding and the corresponding GP hyperparameters.

For this, we prefer the active subspace to be learned (and not random, which usually requires several random AS to work well, in practice and theoretically, see Cartis et al. (2023)). There we follow Wycoff et al. (2021), because the required number of hyperparameters to learn the linear embedding remains limited compared to say, Garnett et al. (2014); Letham et al. (2020). Following the modularization principle Liu et al. (2009), that is, separating inference of different modules, we chose to perform a two-stage approach, rather than the single-stage one described in Appendix A. In Step 8, first a tensor product high-dimensional GP model is trained on the residuals between the observations and predictions by the coarse

model from Step 7. From this model, an estimation of the active subspace matrix $\mathbf{C}$ is obtained (Step 9), following Wycoff et al. (2021). The eigen vectors $\mathbf{U}$ of $\mathbf{C}^{(n)} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ provide the new coordinate system, i.e., with rotated coordinates $\mathbf{X}_{E,r} = \mathbf{X}_E\mathbf{U}_{1:r}$ (assuming that $\mathbf{X}$ is centered). It remains to select the number r of eigen vectors from $\mathbf{U}$. Since r is a discrete parameter, one simple workaround is to optimize the lengthscale parameters for various values of r, before selecting the best overall value (Step 10). One can consider that lengthscales for the inactive dimensions are set to infinity, such that this remains the same model defined on the full rotated space. Note that this can result in a noisy model, where the noise subsumes the contributions of the inactive dimensions.

4 Empirical Evaluation

We conduct a comparative analysis between the multi-fidelity approach and baseline methods on synthetic functions and datasets. Rather than taking very large input dimensions d and data sizes n, we focus on the low data regime, considering n up to 500 and d up to 32.

4.1 Setup

As a baseline, we use a standard GP model (hereafter denoted by Ref), with a tensor product kernel. The `R` R Core Team (2023) package `hetGP` Binois & Gramacy (2021) is used for learning of the hyperparameters, where the initialization of the hyperparameters is complemented by an initialization with the `R` package `RobustGaSP` Gu et al. (2022) for a robust hyperparameter estimation Gu et al. (2018). We entertain an additional variant of standard GPs, with an isotropic kernel (Iso). Additionally we assess the individual component models of the multi-fidelity approach: a first order additive model (Add) and linearly embedded GP (AS). A multi-fidelity model with a standard GP for the finer level is also entertained (MF), in addition to the version with active subspace (ASMF, which is our main proposal). We further add naive variants (n-) of the multi-fidelity models, involving a direct summation of the additive model with the one on the residuals. The implementation of the proposed models is in the Supplementary Material to reproduce the results. All use Matérn 5/2 kernels in these experiments.

For test functions, we start with draws from GPs with $d = 8, 15$, avoiding model mismatch. That is, we consider draws from standard GPs and first order additive GPs. The subsequent set of tests is with classical toy problems: Sobol ($d = 8$) Marrel et al. (2009), penicillin ($d = 7$) Liang & Lai (2021), Levy ($d = 10, 20$) Laguna & Marti (2005) and Cola

($d = 17$) Mathar & Zilinskas (1994). We also embed lower dimensional test functions, Hartmann3 ($d_e = 3$) with a random AS matrix with $d = 8, 15$ and Branin ($d_e = 2$) Dixon (1978) with a random hashing matrix with $d = 10$. We complement these by adding an additive GP realization to the linearly embedded Hartmann3 function. From a thousand randomly sampled locations where these benchmarks are evaluated, a training set is extracted. Lastly, we use real datasets `BostonHousing` ($d = 13$), `Concrete` ($d = 8$) Newman et al. (1998), `pumadyn` ($d = 8, 32$) Corke (1996) and `CASP` ($d = 9$) Rossi & Ahmed (2015). All test sets are centered, and rescaled to unit variance. As for metrics, we rely on the root mean square error (RMSE) and score (or log-predictive density, Gneiting & Raftery, 2007), computed on the remaining data after training.

4.2 Results

The results are presented for the RMSE (lower is better) and score (higher is better) in Figures 2 and 3. We threshold scores at -5 for better visualization. Before delving into specific details, the multi-fidelity plus AS is in general at least as well as regular GPs. Notably, it can improve significantly the results when additive or low intrinsic dimensionality is present. The few exceptions, e.g., on standard GP samples, occur mostly for the lowest budgets and with a small difference. It is noteworthy that regular GPs are in general not worse than most alternatives, and in particular when first-order additive or AS-alone structure are not present. This underscores the importance of meticulous hyperparameter tuning for high-dimensional GPs, where larger values can be taken to offset greater distances, without resulting in conditioning issues of the covariance matrix.

The naive and AR multi-fidelity models seem to perform similarly on the RMSE, but the AR model generally yields better scores, both with full and AS kernels. Perhaps unsurprisingly, the best results of the MF models are obtained when the additive model performs well: e.g., for Sobol, penicilin, addGP. Then the AS version of the MF GPs tends to outperform the full GP one, where the best results are obtained when AS structure is present as well: penicillin, puma, addGP + Hartman or Levy. This improvement can be attributed to either truncating the number of active dimensions, or simply to the rotation applied on the input space when keeping all variables. This can be seen in Appendix in Figure 4 (also note that the selected dimension is usually larger than the actual embedding one, which is recommended in practice Letham et al. (2020) and theoretically Cartis & Otemissov (2022)). While additive and AS models excel when such structure is present, they tend to perform poorly when this assumption does not hold, especially in terms of score. This effect is more

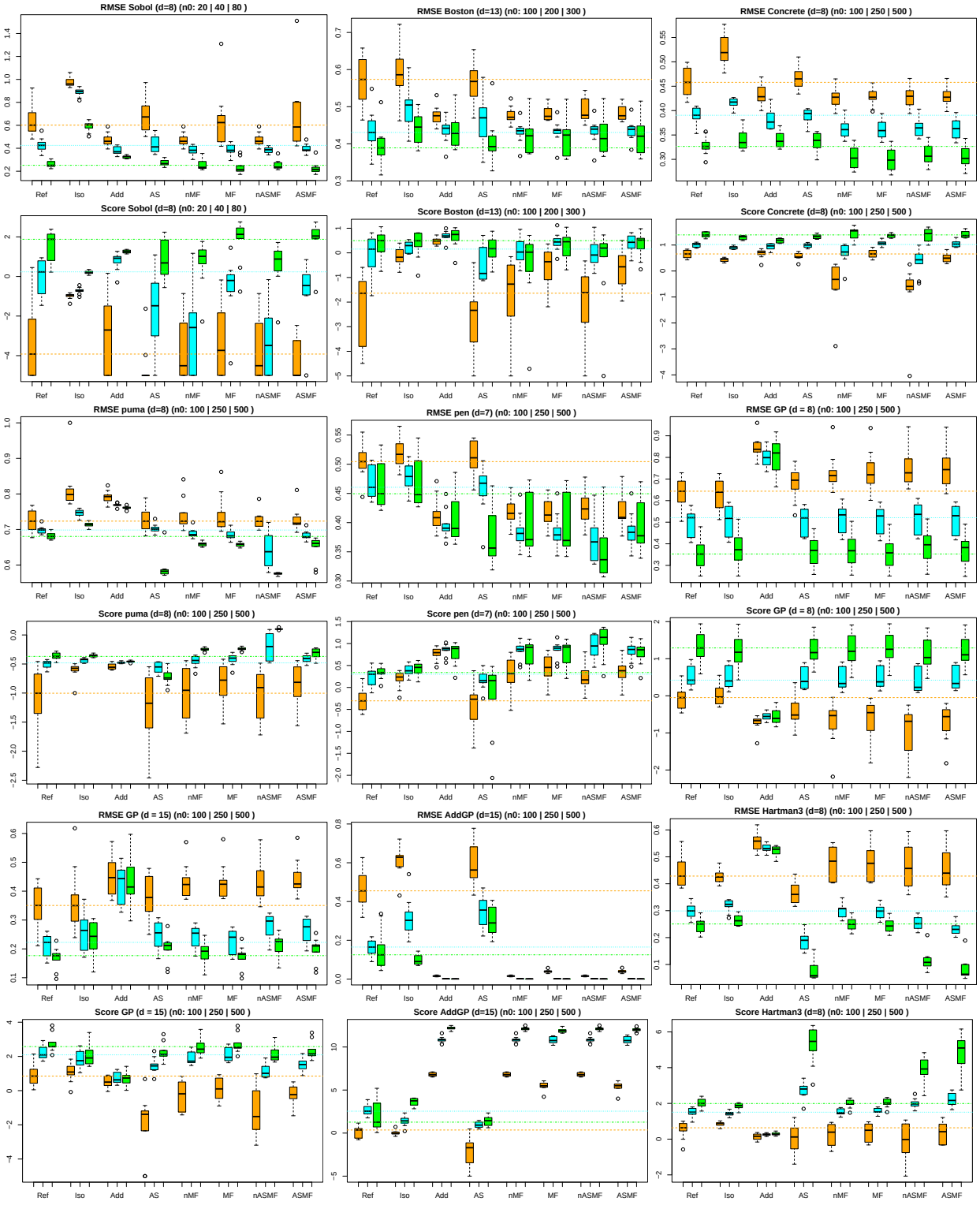

Figure 2: First part RMSE and score results. The color lines indicate the baseline result from standard GP models.

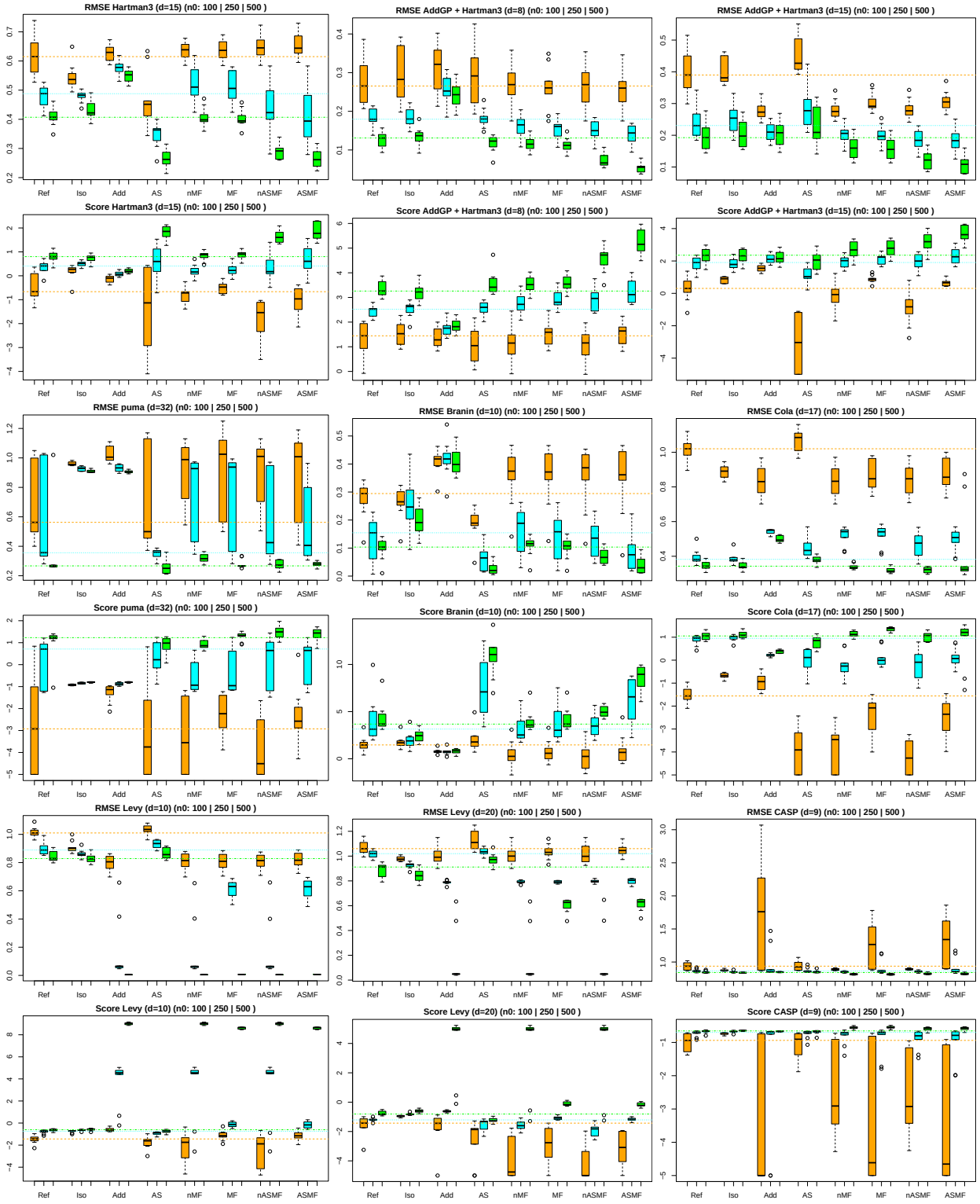

Figure 3: Second part RMSE and score results. The color lines indicate the baseline result from standard GP models.

pronounced in additive models, as AS GP ones can compensate using $r = d$ dimensions, ultimately not reducing the dimension.

Then for problems with simultaneously additive and AS structures, the dedicated GP models perform best. Isotropic GPs may make a reasonable initial choice with low data, as they are much easier to infer, but they quickly become less effective than anisotropic ones. Timings are provided in Appendix C, Figure 6, where the repeated optimization of the likelihood to estimate the best low dimension r in the AS models shows. Considering multi-fidelity does not add much computational effort. In terms of budget, more data is beneficial to all models, as reflected by the larger boxplots for lower budgets. The effect of the budget is the most striking on the AS models, suggesting that a minimal amount of data is essential for robust inference. Conversely, the additive model exhibits the least change with increasing budget and can perform well even with lowest budgets.

Finally, we include an indicative comparison with the higher-order additive model (OAK) from Lu et al. (2022), as depicted in Figure 5 in Appendix C. However, the results are harder to interpret: the predictive variance is not returned to compute scores, additional scalings are performed, plus a measure on the input space is needed. OAK can perform better than the alternatives on some test cases, but worse on the cases with active subspaces. This highlights the difficulty of learning high-order interactions in additive models.

5 Conclusion and Perspectives

We propose a simple solution to properly combine the predominant structural assumptions for high-dimensional modeling: additivity and low intrinsic dimensionality. The resulting multi-fidelity model is simple to construct and robust to incorrect assumptions. The promising results obtained open perspectives in several main directions. First, the inference of GP hyperparameters for high-dimensional problems may be improved, potentially starting with GPs that have pre-selected lengthscales, as suggested in Appriou et al. (2023). Then a comparison could be conducted with direct inference of the active subspaces matrix within GPs, see e.g., Letham et al. (2020) or Garnett et al. (2014). Given that larger datasets may be necessary for more precise inference of such features, a combination with sparse GP models, e.g., as in Moss et al. (2023), would be considered. This could further include input and output warpings, as already advocated by Lin & Joseph (2020); Lu et al. (2022). A second direction to explore is the use of less linear multi-fidelity models as summarized by Brevault et al. (2020), or even multi source models, see, e.g., Poloczek et al. (2017). This would be

beneficial when combining models whose structural assumptions have no natural ordering, like BOCK Oh et al. (2018), and higher order additive models Lu et al. (2022). Non-linear dimension reduction is another appealing candidate, see e.g., Guhaniyogi & Dunson (2016). Lastly, GP modeling shines in sequential procedures, where existing works only focus on individual aspects, say additivity Schwabe (1995), multi-fidelity Le Gratiet & Cannamela (2015) or active subspace estimation Wycoff et al. (2021). Future research could delve into the alignment of these goals compared to Bayesian optimization, exploring how these aspects synergize in sequential decision-making processes.

References

Appriou, T., Rullière, D., and Gaudrie, D. Combination of optimization-free kriging models for high-dimensional problems. *Computational Statistics*, pp. 1–23, 2023.

Binois, M. and Gramacy, R. B. hetGP: Heteroskedastic Gaussian process modeling and sequential design in R. *Journal of Statistical Software*, 98(13):1–44, 2021. doi: 10.18637/jss.v098.i13.

Binois, M. and Wycoff, N. A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization. *ACM Transactions on Evolutionary Learning and Optimization*, 2(2):1–26, 2022.

Bouhlel, M. A., Bartoli, N., Otsmane, A., and Morlier, J. Improving kriging surrogates of high-dimensional design models by partial least squares dimension reduction. *Structural and Multidisciplinary Optimization*, 53(5):935–952, 2016.

Brevault, L., Balesdent, M., and Hebbal, A. Overview of Gaussian process based multi-fidelity techniques with variable relationship between fidelities, application to aerospace systems. *Aerospace Science and Technology*, 107:106339, 2020.

Cao, J., Guinness, J., Genton, M. G., and Katzfuss, M. Scalable Gaussian-process regression and variable selection using Vecchia approximations. *The Journal of Machine Learning Research*, 23(1):15799–15828, 2022.

Carpentier, A. and Munos, R. Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. In *Artificial Intelligence and Statistics*, pp. 190–198, 2012.

Cartis, C. and Otemissov, A. A dimensionality reduction technique for unconstrained global optimization of functions with low effective dimensionality. *Information and Inference: A Journal of the IMA*, 11(1):167–201, 2022.

Cartis, C., Massart, E., and Otemissov, A. Bound-constrained global optimization of functions with low effective dimensionality using multiple random embeddings. *Mathematical Programming*, 198(1):997–1058, 2023.

Constantine, P. G. *Active subspaces: Emerging ideas for dimension reduction in parameter studies*. SIAM, 2015.

Constantine, P. G., Dow, E., and Wang, Q. Active subspace methods in theory and practice: applications to kriging surfaces. *SIAM Journal on Scientific Computing*, 36(4):A1500–A1524, 2014.

Corke, P. I. A robotics toolbox for Matlab. *IEEE Robotics & Automation Magazine*, 3(1): 24–32, 1996.

Dixon, L. C. W. The global optimization problem: an introduction. *Towards Global Optimiation 2*, pp. 1–15, 1978.

Djolonga, J., Krause, A., and Cevher, V. High-dimensional Gaussian process bandits. In *Neural Information Processing Systems*, pp. 1025–1033, 2013.

Durrande, N., Ginsbourger, D., and Roustant, O. Additive kernels for Gaussian process modeling. *Annales de la Facultée de Sciences de Toulouse*, pp. 17, 2012.

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.

Duvenaud, D. K., Nickisch, H., and Rasmussen, C. E. Additive Gaussian processes. In *Advances in neural information processing systems*, pp. 226–234, 2011.

Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., and Poloczek, M. Scalable global optimization via local Bayesian optimization. *Advances in Neural Information Processing Systems*, 32:5496–5507, 2019.

Forrester, A., Sobester, A., and Keane, A. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

Gardner, J., Guo, C., Weinberger, K., Garnett, R., and Grosse, R. Discovering and exploiting additive structure for Bayesian optimization. In *Artificial Intelligence and Statistics*, pp. 1311–1319, 2017.

Garnett, R. *Bayesian optimization*. Cambridge University Press, 2023.

Garnett, R., Osborne, M. A., and Hennig, P. Active learning of linear embeddings for Gaussian processes. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pp. 230–239. AUAI Press, 2014.

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.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Gramacy, R. B. *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. CRC Press, 2020.

Gu, M., Wang, X., and Berger, J. O. Robust Gaussian stochastic process emulation. *The Annals of Statistics*, 46(6A):3038–3066, 2018.

Gu, M., Palomo, J., and Berger, J. *RobustGaSP: Robust Gaussian Stochastic Process Emulation*, 2022. URL `https://CRAN.R-project.org/package=RobustGaSP`. R package version 0.6.5.

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

Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian Processes for Big Data. *Uncertainty in Artificial Intelligence*, 2013.

Journel, A. G. and Rossi, M. When do we need a trend model in kriging? *Mathematical Geology*, 21(7):715–739, 1989.

Kandasamy, K., Schneider, J., and Póczos, B. High dimensional Bayesian optimisation and bandits via additive models. In *International conference on machine learning*, pp. 295–304. PMLR, 2015.

Kennedy, M. C. and O'Hagan, A. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

Laguna, M. and Marti, R. Experimental testing of advanced scatter search designs for global optimization of multimodal functions. *Journal of Global Optimization*, 33:235–255, 2005.

Le Gratiet, L. and Cannamela, C. Cokriging-based sequential design strategies using fast cross-validation techniques for multi-fidelity computer codes. *Technometrics*, 57(3):418–427, 2015.

Le Gratiet, L. and Garnier, J. Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. *International Journal for Uncertainty Quantification*, 4(5), 2014.

Lenz, N. Additivity and Ortho-Additivity in Gaussian Random Fields. Technical report, University of Bern, 2013. URL `https://hal.science/hal-01063741`.

Letham, B., Calandra, R., Rai, A., and Bakshy, E. Re-examining linear embeddings for high-dimensional Bayesian optimization. *Advances in Neural Information Processing Systems*, 33, 2020.

Liang, Q. and Lai, L. Scalable Bayesian optimization accelerates process optimization of penicillin production. In *NeurIPS 2021 AI for Science Workshop*, 2021.

Lin, L.-H. and Joseph, R. V. Transformation and additivity in Gaussian processes. *Technometrics*, 62(4):525–535, 2020.

Liu, F., Bayarri, M., Berger, J., et al. Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis*, 4(1):119–150, 2009.

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

Malu, M., Dasarathy, G., and Spanias, A. Bayesian optimization in high-dimensional spaces: A brief survey. In *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pp. 1–8. IEEE, 2021.

Marrel, A., Iooss, B., Laurent, B., and Roustant, O. Calculations of Sobol indices for the Gaussian process metamodel. *Reliability Engineering & System Safety*, 94(3):742–751, 2009.

Mathar, R. and Zilinskas, A. A class of test functions for global optimization. *Journal of Global Optimization*, 5(2):195–199, 1994.

Moss, H. B., Ober, S. W., and Picheny, V. Inducing point allocation for sparse Gaussian processes in high-throughput Bayesian optimisation. In *International Conference on Artificial Intelligence and Statistics*, pp. 5213–5230. PMLR, 2023.

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.

Nayebi, A., Munteanu, A., and Poloczek, M. A framework for Bayesian optimization in embedded subspaces. In *International Conference on Machine Learning*, pp. 4752–4761, 2019.

Neal, R. M. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. *arXiv preprint physics/9701026*, 1997.

Newman, D., Hettich, S., Blake, C., and Merz, C. UCI repository of machine learning databases, 1998. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Oh, C., Gavves, E., and Welling, M. Bock: Bayesian optimization with cylindrical kernels. In *International Conference on Machine Learning*, pp. 3868–3877. PMLR, 2018.

Petersen, K. B., Pedersen, M. S., et al. The matrix cookbook. *Technical University of Denmark*, 7:15, 2008.

Plate, T. A. Accuracy versus interpretability in flexible modeling: Implementing a tradeoff using Gaussian process models. *Behaviormetrika*, 26(1):29–50, 1999.

Poloczek, M., Wang, J., and Frazier, P. Multi-information source optimization. *Advances in neural information processing systems*, 30, 2017.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2023. URL `https://www.R-project.org/`.

Raponi, E., Wang, H., Bujny, M., Boria, S., and Doerr, C. High dimensional Bayesian optimization assisted by principal component analysis. In *International Conference on Parallel Problem Solving from Nature*, pp. 169–183. Springer, 2020.

Rolland, P., Scarlett, J., Bogunovic, I., and Cevher, V. High-dimensional Bayesian optimization via additive models with overlapping groups. In *International Conference on Artificial Intelligence and Statistics*, pp. 298–307, 2018.

Rossi, R. A. and Ahmed, N. K. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. URL `https://networkrepository.com`.

Sacher, M., Le Maitre, O., Duvigneau, R., Hauville, F., Durand, M., and Lothodé, C. A non-nested infilling strategy for multifidelity based efficient global optimization. *International Journal for Uncertainty Quantification*, 11(1), 2021.

Schwabe, R. Designing experiments for additive nonlinear models. In *MODA4—Advances in Model-Oriented Data Analysis: Proceedings of the 4th International Workshop in Spetses, Greece June 5–9, 1995*, pp. 77–85. Springer, 1995.

Tighineanu, P., Skubch, K., Baireuther, P., Reiss, A., Berkenkamp, F., and Vinogradska, J. Transfer learning with Gaussian processes for Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 6152–6181. PMLR, 2022.

Titsias, M. Variational Learning of Inducing Variables in Sparse Gaussian Processes. *Artificial Intelligence and Statistics*, 2009.

Tripathy, R., Bilionis, I., and Gonzalez, M. Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation. *Journal of Computational Physics*, 321:191 – 223, 2016. ISSN 0021-9991.

Wang, Z., Zoghi, M., Hutter, F., Matheson, D., and De Freitas, N. Bayesian optimization in high dimensions via random embeddings. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pp. 1778–1784, 2013.

Williams, C. K. and Rasmussen, C. E. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. Deep kernel learning. In Gretton, A. and Robert, C. C. (eds.), *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 370–378. PMLR, 2016.

Wycoff, N., Binois, M., and Wild, S. M. Sequential learning of active subspaces. *Journal of Computational and Graphical Statistics*, 30(4):1224–1237, 2021.

Wycoff, N., Binois, M., and Gramacy, R. B. Sensitivity prewarping for local surrogate modeling. *Technometrics*, 64(4):535–547, 2022.

A One Shot Active Subspace Gaussian Process Learning

In practice, $\mathbf{C} = \int_{\mathbf{X}} \nabla(f(\mathbf{x}))^\top \nabla(f(\mathbf{x}))\lambda(d\mathbf{x})$ is often estimated by Monte Carlo, either directly on f when the gradient is available, or on its surrogate. In Wycoff et al. (2021), this AS matrix is expressed in closed form for a GP, which can be further used to reduce the dimension. Let k be a twice differentiable kernel, with derivatives κ, $\mathbf{W}_{i,j} := \int_{\mathcal{X}} \boldsymbol{\kappa}_i(X)^\top \boldsymbol{\kappa}_j(X) d\lambda$, and $E_{i,j} := \int_{\mathcal{X}} \frac{\partial^2 k(X,X)}{\partial x_i \partial x_j} d\lambda$. Then, $C^{(n)}_{i,j} = E_{i,j} - tr\left(\mathbf{K}_n^{-1}\mathbf{W}_{i,j}\right) + \mathbf{y}^\top \mathbf{K}_n^{-1}\mathbf{W}_{i,j}\mathbf{K}_n^{-1}\mathbf{y}$. As a result, estimating a GP with reduced dimension is possible via a two-step process: first fit an high-dimensional GP, then use the corresponding AS matrix to learn a low dimensional GP on the projected data, with kernel $k(\mathbf{x}, \mathbf{x}') = \tilde{k}(\mathbf{A}^\top \mathbf{x}, \mathbf{A}^\top \mathbf{x}')$, assuming a centered $\mathbf{X}$, where $\mathbf{A} = \mathbf{U}_r$, the first r eigen vectors of $\mathbf{C}^{(n)}$.

Nevertheless, the AS matrix $\mathbf{C}^{(n)}$ in fact only depends on the d lengthscale hyperparameters, and the data. What we propose here is to use the same parameterization of the AS matrix of the GP, but learn the parameters via the likelihood of a low dimensional GP, i.e., learn all hyperparameters: $l_1, \ldots, l_r, \theta_1, \ldots \theta_d$ at once, where the l_i (resp. θ_i) are the low (high) dimensional GP lengthscales.

We rely on the work made previously with the derivative of an AS kernel in Wycoff et al. (2021), and give the additional required expressions, that is $\frac{\partial \mathbf{C}}{\partial \theta_i}$:

$$\frac{\partial C^{(n)}_{i,j}}{\partial \theta_i} = \frac{\partial E_{i,j} - tr\left(\mathbf{K}_n^{-1}\mathbf{W}_{i,j}\right) + \mathbf{y}^\top \mathbf{K}_n^{-1}\mathbf{W}_{i,j}\mathbf{K}_n^{-1}\mathbf{y}}{\partial \theta_i}.$$

Hence we need $\frac{\partial E_{i,j}}{\partial \theta_i}$, $\frac{\partial \mathbf{K}_n^{-1}}{\partial \theta_i}$ and $\frac{\partial \mathbf{W}_{i,j}}{\partial \theta_i}$. These are combined to get:

$$\frac{\partial C^{(n)}_{i,j}}{\partial \theta_i} = \frac{\partial E_{i,j}}{\partial \theta_i} - tr\left(\frac{\partial \mathbf{K}_n^{-1}}{\partial \theta_i}\mathbf{W}_{i,j} + \mathbf{K}_n^{-1}\frac{\partial \mathbf{W}_{i,j}}{\partial \theta_i}\right) + \mathbf{y}^\top \frac{\partial \mathbf{K}_n^{-1}}{\partial \theta_i}\mathbf{W}_{i,j}\mathbf{K}_n^{-1}\mathbf{y} + \mathbf{y}^\top \mathbf{K}_n^{-1}\frac{\partial \mathbf{W}_{i,j}}{\partial \theta_i}\mathbf{K}_n^{-1}\mathbf{y} + \mathbf{y}^\top \mathbf{K}_n^{-1}\mathbf{W}_{i,j}\frac{\partial \mathbf{K}_n^{-1}}{\partial \theta_i}\mathbf{y}.$$

and up to the likelihood level:

$$\frac{\partial \log L}{\partial \theta_i} = \frac{\partial\left(const. - \frac{n}{2}\log \mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}|\right)}{\partial \theta_i} = \frac{n}{2\hat{\sigma}^2}\mathbf{y}^\top \mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta_i}\mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}Tr\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta_i}\right)$$

where, using the chain rule $\frac{\partial K_{i,j}}{\partial \theta_i} = \frac{\partial K_{i,j}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \theta_i}$. More precisely, using Petersen et al. (2008), involving the eigen vectors $\mathbf{U}_l$ and corresponding eigen value λ_l of $\mathbf{C}$, and pseudo-inverses $\dagger$:

$$\frac{\partial \mathbf{U}_l}{\partial \theta_i} = (\lambda_l \mathbf{I} - \mathbf{C})^\dagger \frac{\partial \mathbf{C}}{\partial \theta_i} \mathbf{U}_l.$$

As an example, for a Gaussian kernel in the reduced dimension too, $\mathbf{h} = (\mathbf{x}_i - \mathbf{x}_j)$, such that: $\frac{\partial K_{i,j}}{\partial \mathbf{W}} = 2Diag(\boldsymbol{l})\mathbf{W}\mathbf{h}\mathbf{h}^\top K_{i,j}$.

For this Gaussian kernel case, parameterized by $k(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^{d} \exp\left(-\left(\frac{x_i - x_i'}{\theta_i}\right)^2\right)$ (denoting $a = x_i$, $b = x_i'$ and $t = \theta_i$):

$\frac{\partial E_{i,j}}{\partial \theta} = \frac{\partial}{\partial \theta} \frac{\delta_{i=j}}{\theta^2} = -2\delta_{i=j}\theta^{-3}$

$$\frac{\partial w_{i,i}(a,b,t)}{\partial t} = -\frac{\left(2t\left((b+a-2)\left(2t^2-(b+a-2)^2\right)\mathrm{e}^{\frac{b+a}{t^2}} - (b+a)\left(2t^2-(b+a)^2\right)\mathrm{e}^{\frac{1}{t^2}}\right)\mathrm{e}^{\frac{-b^2-a^2-2}{2t^2}}\right)}{16t^6}$$
$$-\frac{\left(\sqrt{\pi}\left(\mathrm{erf}\left(\frac{b+a}{2t}\right) - \mathrm{erf}\left(\frac{b+a-2}{2t}\right)\right)\left(2t^2-2bt+2at-b^2+2ab-a^2\right)\left(2t^2+2bt-2at-b^2+2ab-a^2\right)\mathrm{e}^{\frac{-b^2+2ab-a^2-1}{4t^2}}\right)}{16t^6}$$

$$\frac{\partial w_{i,j}(a,b,t)}{\partial t} = \frac{-2\left(a-b\right)t\mathrm{e}^{-\frac{(a-b)^2}{4t^2}}\left(\left(b+a\right)\mathrm{e}^{-\frac{(b+a)^2}{4t^2}} - \left(b+a-2\right)\mathrm{e}^{-\frac{(b+a-2)^2}{4t^2}}\right)}{8t^4}$$
$$+\frac{4t\left(-\left(b^2+a^2\right)\mathrm{e}^{-\frac{b^2+a^2}{2t^2}} - \left(-b^2+2\left(b+a-1\right)-a^2\right)\mathrm{e}^{\frac{-b^2+2(b+a-1)-a^2}{2t^2}}\right)}{8t^4}$$
$$+\frac{2\sqrt{\pi}\left(a-b\right)\left(\mathrm{erf}\left(\frac{b+a-2}{2t}\right) - \mathrm{erf}\left(\frac{b+a}{2t}\right)\right)t^2\mathrm{e}^{-\frac{(a-b)^2}{4t^2}} - \sqrt{\pi}\left(a-b\right)^3\left(\mathrm{erf}\left(\frac{b+a-2}{2t}\right) - \mathrm{erf}\left(\frac{b+a}{2t}\right)\right)\mathrm{e}^{-\frac{(a-b)^2}{4t^2}}}{8t^4}$$

$$\frac{\partial I_{l,l}(a,b,t)}{\partial t} = \frac{2\sqrt{\pi}t^2\mathrm{e}^{-\frac{(b-a)^2}{4t^2}}\left(\mathrm{erf}\left(\frac{b+a}{2|t|}\right) - \mathrm{erf}\left(\frac{b+a-2}{2|t|}\right)\right) + \sqrt{\pi}\left(b-a\right)^2\mathrm{e}^{-\frac{(b-a)^2}{4t^2}}\left(\mathrm{erf}\left(\frac{b+a}{2|t|}\right) - \mathrm{erf}\left(\frac{b+a-2}{2|t|}\right)\right)}{4t^2}$$
$$+\frac{2\mathrm{e}^{-\frac{(b-a)^2}{4t^2}}\left(\left(b+a-2\right)\mathrm{e}^{-\frac{(b+a-2)^2}{4t^2}} - \left(b+a\right)\mathrm{e}^{-\frac{(b+a)^2}{4t^2}}\right)}{4t}$$

B Complements on the Auto-regressive Multi-fidelity Model

For the implementation, we start by giving some log-likelihood derivatives. Then we discuss the link to the recursive formulation of the AR multi-fidelity model.

B.1 Log-likelihood Derivatives

Derivatives of the log-likelihood are given, e.g., in Forrester et al. (2008). The only change is the log-likelihood derivative that requires adaptation in the additive case.

For the coarse level model with an additive kernel where $\mathbf{K} = \sum_{i=1}^{d} \alpha_i \mathbf{K}^{(i)}(\theta_i) + g\mathbf{I}$:

$$L = -n/2 \log(2\pi) - 1/2 \mathbf{y}_C^\top \mathbf{K}^{-1} \mathbf{y}_C - 1/2 \log |\mathbf{K}|$$

$$\frac{\partial L}{\partial \theta_i} = 1/2 \mathbf{y}_C^\top \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \theta_i} \mathbf{K}^{-1} \mathbf{y}_C - 1/2 Tr\left(\mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \theta_i}\right)$$

$$\frac{\partial L}{\partial \alpha_i} = 1/2 \mathbf{y}_C^\top \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \alpha_i} \mathbf{K}^{-1} \mathbf{y}_C - 1/2 Tr\left(\mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \alpha_i}\right) = 1/2 \mathbf{y}_C^\top \mathbf{K}^{-1} \mathbf{K}^{(i)} \mathbf{K}^{-1} \mathbf{y}_C - 1/2 Tr\left(\mathbf{K}^{-1} \mathbf{K}^{(i)}\right)$$

Then for the AR multi-fidelity kernel:

$$\frac{\partial \tilde{L}}{\partial \rho} = \frac{\partial - n/2 \log(\hat{\sigma}_d^2)}{\partial \rho} = -1/2 \frac{\partial (\mathbf{y}_E - \rho \mathbf{y}_C)^\top \mathbf{K}^{-1} (\mathbf{y}_E - \rho \mathbf{y}_C)}{\partial \theta_i} = \mathbf{y}_C \mathbf{K}^{-1} (\mathbf{y}_E - \rho \mathbf{y}_C)$$

B.2 Recursive Formulation

Le Gratiet & Garnier (2014) provide a recursive formulation of the multi-fidelity AR model, which is equivalent to the one by Kennedy & O'Hagan (2000) but only in the deterministic case. This formulation writes:

$$\begin{aligned} m_{n,E}(\mathbf{x}) &= m_{n,C}(\mathbf{x}) + k_E(\mathbf{x}, \mathbf{X}_E) \mathbf{K}_E^{-1} (\mathbf{y}^{(E)} - \rho \mathbf{y}^{(C)}), \\ s_{n,E}^2(\mathbf{x}) &= \rho^2 s_{n,C}(\mathbf{x}) + k_E(\mathbf{x}, \mathbf{x}) - k_E(\mathbf{x}, \mathbf{X}_E) \mathbf{K}_E^{-1} k_E(\mathbf{X}_E, \mathbf{x}) \end{aligned} \tag{3}$$

which reduces the computational complexity of fitting the finer level and gives the predictive quantities at all fidelity levels. We give here a simple proof of the equivalence in this case, then show that it does not apply in the noisy one.

B.2.1 Deterministic Case

In the deterministic case, when both designs are equal, $\mathbf{X}_C = \mathbf{X}_E$ (of size n):

$$\tilde{\mathbf{K}} = \begin{bmatrix} k_C(\mathbf{X}_C, \mathbf{X}_C) & \rho k_C(\mathbf{X}_C, \mathbf{X}_E) \\ \rho k_C(\mathbf{X}_E, \mathbf{X}_C) & \rho^2 k_C(\mathbf{X}_E, \mathbf{X}_E) + k_E(\mathbf{X}_E, \mathbf{X}_E) \end{bmatrix} := \begin{bmatrix} \mathbf{K}_C & \rho \mathbf{K}_C \\ \rho \mathbf{K}_C & \rho^2 \mathbf{K}_C + \mathbf{K}_E \end{bmatrix}$$

Similarly, $\tilde{\mathbf{k}}(\mathbf{x}) := [\rho \mathbf{k}_C, \rho^2 \mathbf{k}_C + \mathbf{k}_E]^\top$ for shorter notation (dropping the dependence on $\mathbf{x}$).

Then the block-matrix inverse formula Petersen et al. (2008) gives, following the notations there: $\mathbf{C}_1 = \mathbf{K}_C - \rho^2\mathbf{K}_C(\mathbf{K}_E + \rho^2\mathbf{K}_C)^{-1}\mathbf{K}_C$ and $\mathbf{C}_2 = \rho^2\mathbf{K}_C + \mathbf{K}_E - \rho^2\mathbf{K}_C = \mathbf{K}_E$ where this second equality is used for expressing $\tilde{\mathbf{K}}^{-1}$: $\tilde{\mathbf{K}}^{-1} = \begin{bmatrix} \rho^2\mathbf{K}_E^{-1} + \mathbf{K}_C^{-1} & -\rho\mathbf{K}_E^{-1} \\ -\rho\mathbf{K}_E^{-1} & \mathbf{K}_E^{-1} \end{bmatrix}$.

Consequently, for the predictive equations:

$\tilde{\mathbf{K}}^{-1}\tilde{\mathbf{k}} = [\rho^3\mathbf{K}_E^{-1}\mathbf{k}_C + \rho\mathbf{K}_C^{-1}\mathbf{k}_C - \rho^3\mathbf{K}_E^{-1}\mathbf{k}_C - \rho\mathbf{K}_E^{-1}\mathbf{k}_E, -\rho^2\mathbf{K}_E^{-1}\mathbf{k}_C + \rho^2\mathbf{K}_E^{-1}\mathbf{k}_C + \mathbf{K}_E^{-1}\mathbf{k}_E] = [\rho\mathbf{K}_C^{-1}\mathbf{k}_C - \rho\mathbf{K}_E^{-1}\mathbf{k}_E, \mathbf{K}_E^{-1}\mathbf{k}_E]$

such that

$m_{n,E}(\mathbf{x}) = \tilde{\mathbf{k}}^\top\tilde{\mathbf{K}}^{-1}\tilde{\mathbf{y}} = \rho\mathbf{k}_C\mathbf{K}_C^{-1}\mathbf{y}_C - \rho\mathbf{k}_E\mathbf{K}_E^{-1}\mathbf{y}_C + \mathbf{k}_E\mathbf{K}_E^{-1}\mathbf{y}_E = m_{n,C}(\mathbf{x}) + \mathbf{k}_E\mathbf{K}_E^{-1}(\mathbf{y}_E - \rho\mathbf{y}_C)$

and

$s^2_{n,E}(\mathbf{x}) = \tilde{\mathbf{k}}^\top\tilde{\mathbf{K}}^{-1}\tilde{\mathbf{k}} = \rho^2\mathbf{k}_C\mathbf{K}_C^{-1}\mathbf{k}_C - \rho^2\mathbf{k}_C\mathbf{K}_E^{-1}\mathbf{k}_E + \rho^2\mathbf{k}_C\mathbf{K}_E^{-1}\mathbf{k}_E + \mathbf{k}_E\mathbf{K}_E^{-1}\mathbf{k}_E = \rho^2\mathbf{k}_C\mathbf{K}_C^{-1}\mathbf{k}_C + \mathbf{k}_E\mathbf{K}_E^{-1}\mathbf{k}_E = \rho^2 s^2_{n,C}(\mathbf{x}) + s^2_{n,d}(\mathbf{x})$

For the non equal DoE, $\mathbf{X}_C$ can be split into $[\mathbf{X}_E, \mathbf{X}_R]$ where $\mathbf{X}_R$ are the designs where only the cheap level is evaluated. In this case, we still have that $\mathbf{C}_2 = \rho^2\mathbf{K}_C(\mathbf{X}_E, \mathbf{X}_E) + \mathbf{K}_E - \rho^2\mathbf{K}_C(\mathbf{X}_E, \mathbf{X}_E) = \mathbf{K}_E$. The last term is obtained by realizing that $\mathbf{A}_{12}$ is the first n_E lines of $\mathbf{K}_C(\mathbf{X}_C, \mathbf{X}_C) = \mathbf{A}_{11}$. Then $\mathbf{A}_{12}\mathbf{A}_{11}^{-1} = [\mathbf{I}, \mathbf{0}]$ and $\mathbf{A}_{11}^{-1}\mathbf{A}_{21} = [\mathbf{I}, \mathbf{0}]^\top$ such that:

$$\tilde{\mathbf{K}}^{-1} = \begin{bmatrix} \begin{bmatrix} \rho^2\mathbf{K}_E^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \mathbf{K}_C(\mathbf{X}_C, \mathbf{X}_C)^{-1} & \begin{bmatrix} -\rho\mathbf{K}_E^{-1} \\ \mathbf{0} \end{bmatrix} \\ \begin{bmatrix} -\rho\mathbf{K}_E^{-1} & \mathbf{0} \end{bmatrix} & \mathbf{K}_E^{-1} \end{bmatrix}$$

Hence, with $\tilde{\mathbf{k}} = [\rho k_C(\mathbf{X}_E, \mathbf{x}), \rho k_C(\mathbf{X}_R, \mathbf{x}), \rho^2 k_C(\mathbf{X}_E, \mathbf{x}) + \mathbf{k}_E]$, the same simplifications as above occur, giving the equivalence between the two formulations. From the expression of $\tilde{\mathbf{K}}^{-1}$, only $n \times n$ inverses and determinants need to be computed.

B.2.2 Noisy Coarse Function

Now assume that the low fidelity function is noisy, as is usually the case for an additive model. This time, the application of the block inverse matrix when $\mathbf{X}_E = \mathbf{X}_C$ on:

$$\check{\mathbf{K}} = \begin{bmatrix} k_C(\mathbf{X}_C, \mathbf{X}_C) + g\mathbf{I} & \rho k_C(\mathbf{X}_C, \mathbf{X}_E) \\ \rho k_C(\mathbf{X}_E, \mathbf{X}_C) & \rho^2 k_C(\mathbf{X}_E, \mathbf{X}_E) + k_E(\mathbf{X}_E, \mathbf{X}_E) \end{bmatrix} := \begin{bmatrix} \mathbf{K}_C + \mathbf{D} & \rho\mathbf{K}_C \\ \rho\mathbf{K}_C & \rho^2\mathbf{K}_C + \mathbf{K}_E \end{bmatrix}$$

gives, (following again notations from Petersen et al. (2008)): $\mathbf{C}_1 = (\mathbf{K}_C + \mathbf{D}) - \rho^2\mathbf{K}_C(\mathbf{K}_E + \rho^2\mathbf{K}_C)^{-1}\mathbf{K}_C$ and $\mathbf{C}_2 = \rho^2\mathbf{K}_C + \mathbf{K}_E - \rho^2\mathbf{K}_C(\mathbf{K}_C + \mathbf{D})^{-1}\mathbf{K}_C = \mathbf{K}_E + \rho^2(\mathbf{K}_C^{-1} + g^{-1}\mathbf{I})^{-1} = \mathbf{K}_E + g\rho^2\mathbf{K}_C(\mathbf{K}_C + g\mathbf{I})^{-1}$ using the Woodbury identity. Even though it allows to reduce the computational complexity of the direct multi-fidelity approach, it does not lead to the expressions from the recursive formulation. In particular, the recursive variance

expression does not equal zero at $\mathbf{X}_E$ since the low fidelity variance is greater than zero.

C Additional Results

To complement the results provided in the main part, Figure 4 focuses on the estimation of the low intrinsic dimensionality. Then a comparison on the RMSE for the OAK model by Lu et al. (2022) is given in Figure 5, before general timing results in Figure 6. The experiments have been performed on four 2.40GHz Intel cores.

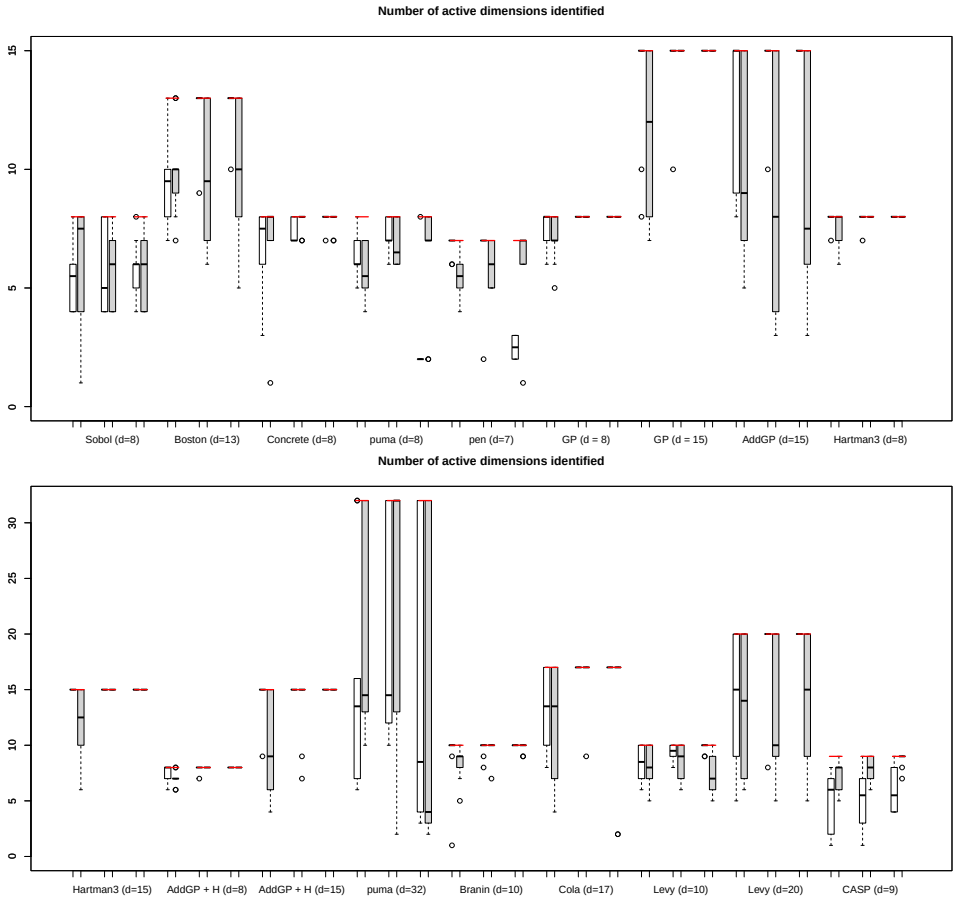

Figure 4: Number of active dimensions kept for the AS GP (left boxplots) and MF AS GP (right gray-filled boxplots). The red segments indicate the number of variables of the problem (d).

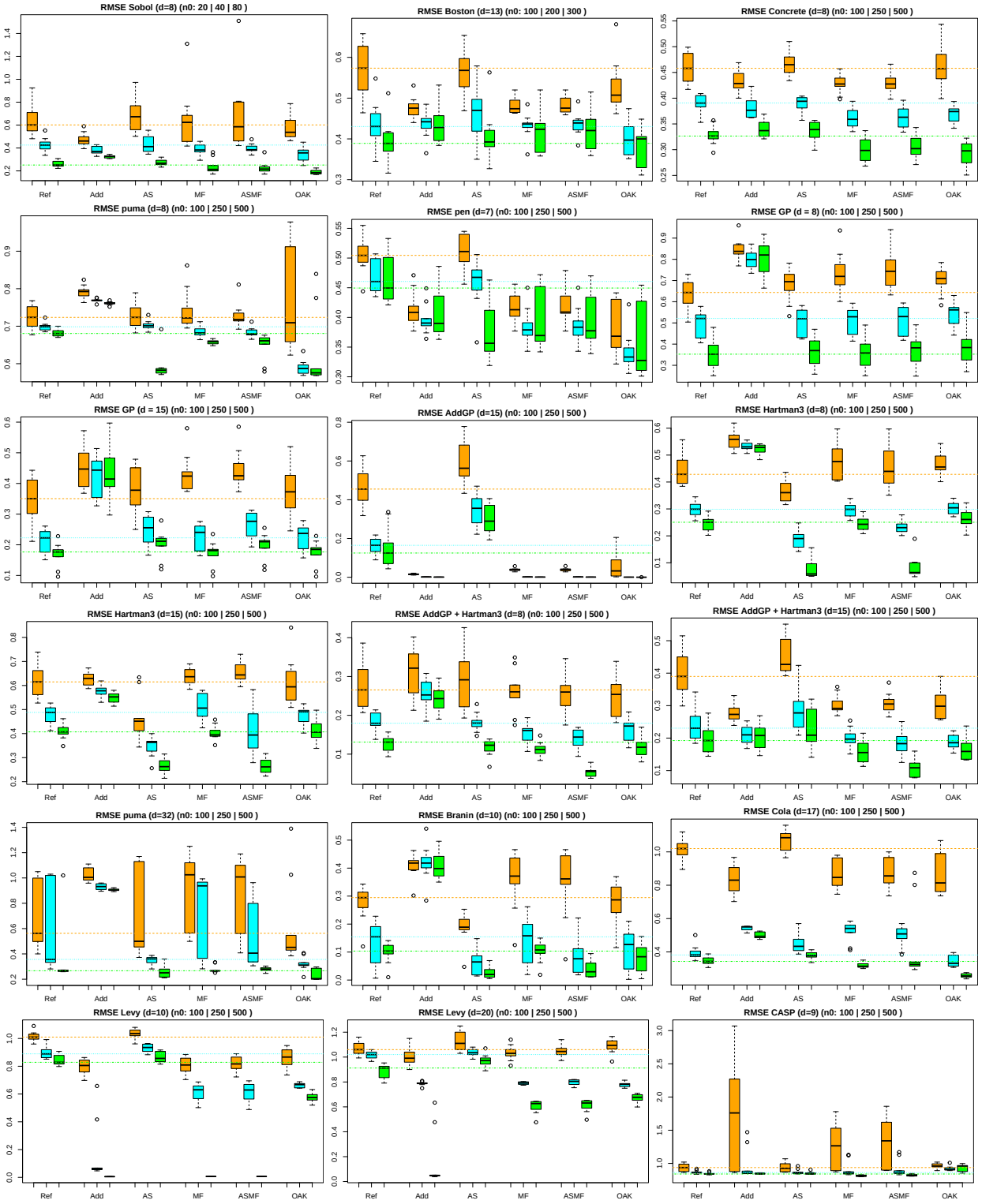

Figure 5: Additional RMSE results including the OAK model. The color lines indicate the baseline result from standard GP models.

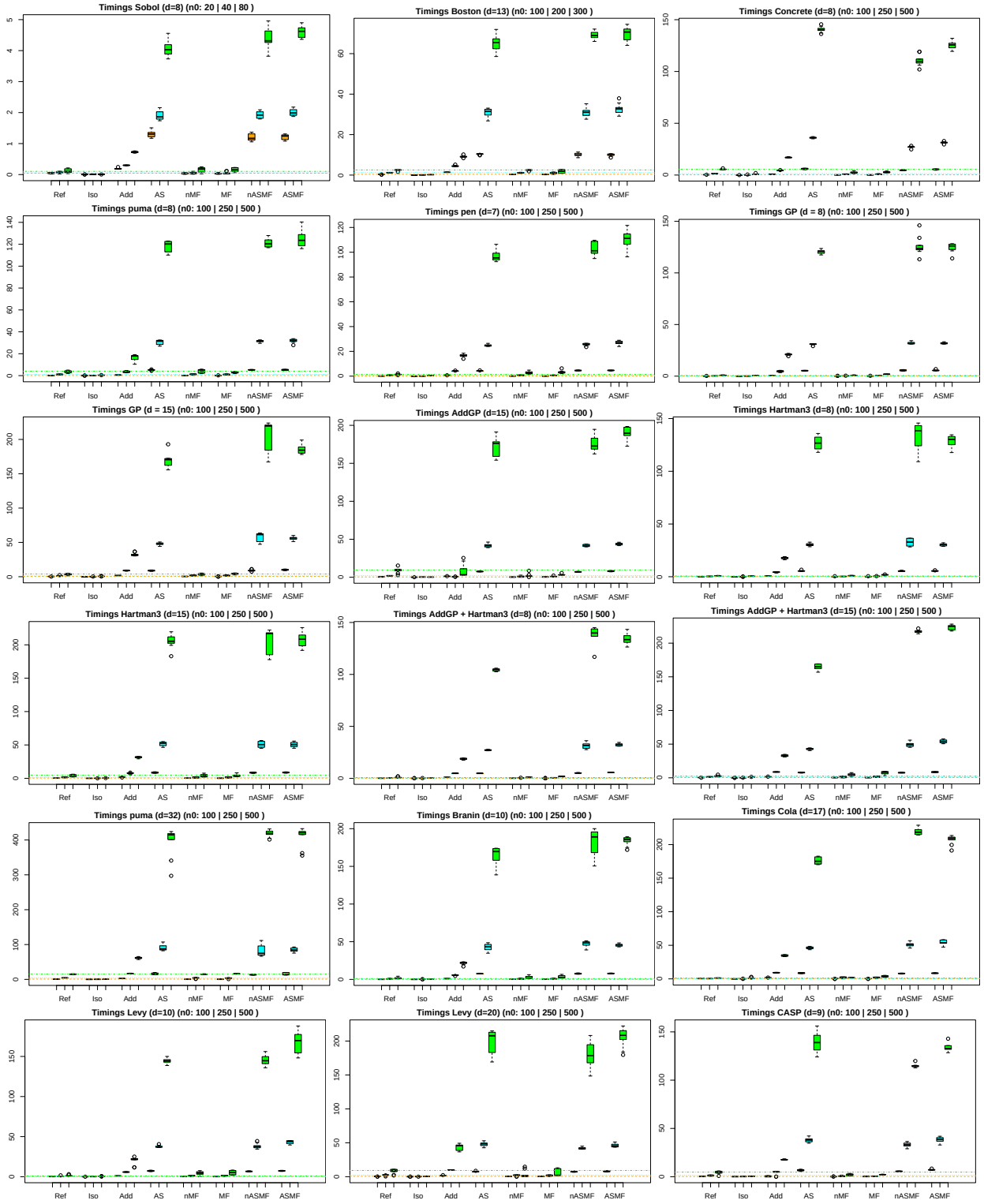

Figure 6: Timings in seconds. The color lines indicate the baseline result from standard GP models.